MySQL数据目录

**章节目录**

目录

[MySQL数据目录 1](#_Toc89631954)

[14.1 数据字典模式 1](#_Toc89631955)

[14.2 移除基于文件的元数据存储 2](#_Toc89631956)

[14.3 词典数据的事务性存储 3](#_Toc89631957)

[14.4 字典对象缓存 3](#_Toc89631958)

[14.5 INFORMATION\_SCHEMA和数据字典集成 4](#_Toc89631959)

[14.6 串行化字典信息（SDI 6](#_Toc89631960)

[14.7 数据字典使用方面的差异 6](#_Toc89631961)

[14.8 数据字典的限制 8](#_Toc89631962)

MySQL服务器集成了一个事务型数据字典，用于存储有关数据库对象的信息。在以前的MySQL版本中，字典数据存储在元数据文件、非事务性表和特定存储引擎的数据字典中。

本章描述了数据字典的主要特征、好处、使用差异和限制。关于数据字典功能的其他含义，请参阅MySQL 8.0发布说明中的 "数据字典说明 "部分。

MySQL数据字典的好处包括

* 集中式数据字典模式的简单性，统一存储字典数据。见第14.1节，"数据字典模式"。
* 消除了基于文件的元数据存储。参见第 14.2 节，"去除基于文件的元数据存储"。
* 对字典数据进行事务性的、防崩溃的存储。参见第 14.3 节，"字典数据的事务性存储"。
* 统一的、集中的字典对象的缓存。参见第 14.4 节，"字典对象的缓存"。
* 对一些INFORMATION\_SCHEMA表进行了更简单和改进的实现。参见第 14.5 节 "INFORMATION\_SCHEMA 和数据字典集成"。
* 原子型DDL。见第13.1.1节，"原子数据定义语句支持"。

重要的是

*与没有数据字典的服务器相比，启用了数据字典的服务器需要一些一般的操作差异；见第14.7节，"数据字典使用差异"。此外，对于升级到MySQL 8.0，升级过程与以前的MySQL版本有些不同，需要你通过检查特定的先决条件来验证你的安装是否准备好升级。欲了解更多信息，请参阅第2.11节 "升级MySQL"，特别是第2.11.5节 "为升级准备你的安装"。*

## 14.1 数据字典模式

数据字典表是受保护的，只能在MySQL的调试构建中访问。然而，MySQL支持通过INFORMATION\_SCHEMA表和SHOW语句访问存储在数据字典表中的数据。关于构成数据字典的表的概述，见数据字典表。

MySQL系统表在MySQL 8.0中仍然存在，可以通过在mysql系统数据库上发出SHOW TABLES语句来查看。一般来说，MySQL数据字典表和系统表之间的区别是，数据字典表包含执行SQL查询所需的元数据，而系统表包含辅助数据，如时区和帮助信息。MySQL系统表和数据字典表在升级方式上也有所不同。MySQL服务器负责管理数据字典的升级。SQL服务器。见数据字典如何升级。升级MySQL系统表需要运行完整的MySQL升级程序。参见第2.11.3节，"MySQL升级过程升级的内容"。

**数据字典是如何升级的**

新版本的MySQL可能包括对数据字典表定义的更改。这种变化存在于新安装的MySQL版本中，但当执行MySQL二进制文件的就地升级时，当使用新的二进制文件重新启动MySQL服务器时，就会应用这些变化。在启动时，服务器的数据字典版本与存储在数据字典中的版本信息进行比较，以确定是否应该升级数据字典表。如果升级是必要的并且被支持，服务器会用更新的定义创建数据字典表，把持久化的元数据复制到新的表中，用新的表原子化地替换旧的表，并重新初始化数据字典。如果不需要升级，则继续启动而不更新数据字典表。

数据字典表的升级是一个原子操作，这意味着所有的数据字典表都会根据需要进行升级，否则操作会失败。如果升级操作失败，服务器启动时就会出现错误。在这种情况下，可以使用旧的服务器二进制文件和旧的数据目录来启动服务器。当再次使用新的服务器二进制文件来启动服务器时，就会重新尝试进行数据字典的升级。

一般来说，在数据字典表成功升级后，不可能用旧的服务器二进制文件重新启动服务器。因此，在数据字典表被升级后，不支持将MySQL服务器二进制文件降级到以前的MySQL版本。

mysqld --no-dd-upgrade选项可以用来阻止启动时自动升级数据字典表。当--no-dd-upgrade被指定时，如果服务器发现服务器的数据字典版本与存储在数据字典中的版本不同，则启动失败，出现一个错误，说明禁止数据字典升级。

**使用MySQL的调试版本查看数据字典表**

数据字典表默认是受保护的，但可以通过编译支持调试的MySQL（使用-DWITH\_DEBUG=1 CMake选项）并指定+d,skip\_dd\_table\_access\_check调试选项和修改器来访问。关于编译调试构建的信息，见第5.9.1.1节，"编译调试的MySQL"。

警告

*不建议直接修改或写入数据字典表，这可能使你的MySQL实例无法运行。*

在用调试支持编译MySQL后，使用这个SET语句使数据字典表对mysql客户会话可见。

mysql> **SET SESSION debug='+d,skip\_dd\_table\_access\_check';**

使用这个查询来检索数据字典表的列表。

mysql> **SELECT name, schema\_id, hidden, type FROM mysql.tables where schema\_id=1 AND hidden='System';**

使用SHOW CREATE TABLE来查看数据字典表的定义。例如

mysql> **SHOW CREATE TABLE mysql.catalogs\G**

## 14.2 移除基于文件的元数据存储

在以前的MySQL版本中，字典数据被部分地存储在元数据文件中。基于文件的元数据存储的问题包括昂贵的文件扫描、易受文件系统相关错误的影响、处理复制和崩溃恢复故障状态的复杂代码，以及缺乏可扩展性，难以为新功能和关系对象添加元数据。

下面列出的元数据文件已从MySQL中删除。除非另有说明，以前存储在元数据文件中的数据现在存储在数据字典表中。

* .frm文件。表元数据文件。随着.frm文件的删除。
  + frm文件结构所规定的64KB的表定义大小限制被取消。
  + INFORMATION\_SCHEMA.TABLES VERSION列报告一个硬编码值10，这是MySQL 5.7中使用的最后一个.frm文件版本。
* .par文件。分区定义文件。InnoDB在MySQL 5.7中停止使用分区定义文件，因为引入了对InnoDB表的本地分区支持。
* .TRN文件。触发器命名空间文件。
* .TRG文件。触发器参数文件。
* .isl文件。InnoDB符号链接文件，包含了在数据目录之外创建的每个表的文件空间的位置。
* db.opt文件。数据库配置文件。这些文件，每个数据库目录一个，包含数据库默认的字符集属性。
* ddl\_log.log文件。该文件包含由数据定义语句产生的元数据操作记录，例如DROP TABLE和ALTER TABLE。

## 14.3 词典数据的事务性存储

数据字典模式在事务性（InnoDB）表中存储字典数据。数据字典表与非数据字典系统表一起位于mysql数据库中。

数据字典表在一个名为mysql.ibd的单一InnoDB表空间中创建，该表空间位于MySQL数据目录中。mysql.ibd表空间文件必须驻留在MySQL数据目录中，其名称不能被修改或被其他表空间使用。

字典数据受到与保护存储在InnoDB表中的用户数据相同的提交、回滚和崩溃恢复能力的保护。

## 14.4 字典对象缓存

字典对象缓存是一个共享的全局缓存，它将以前访问过的数据字典对象存储在内存中，以实现对象的重用，并使磁盘I/O最小化。与MySQL使用的其他缓存机制类似，字典对象缓存使用基于LRU的驱逐策略，从内存中驱逐最近使用的最小对象。

字典对象缓存包括存储不同对象类型的缓存分区。一些缓存分区的大小限制是可配置的，而另一些是硬编码的。

* 表空间定义高速缓存分区。存储表空间定义对象。tablespace\_definition\_cache选项为可以存储在字典对象缓存中的表空间定义对象的数量设置了一个限制。默认值是256。
* 模式定义缓存分区。存储模式定义对象。schema\_definition\_cache选项为可以存储在字典对象缓存中的模式定义对象的数量设置了一个限制。默认值是256。
* 表定义缓存分区。存储表定义对象。对象的限制被设置为max\_connections的值，它的默认值是151。

表定义缓存分区与使用table\_definition\_cache配置选项配置的表定义缓存平行存在。两个缓存都存储表定义，但为MySQL服务器的不同部分服务。一个缓存中的对象不依赖于另一个缓存中对象的存在。

* 存储程序定义缓存分区。存储存储的程序定义对象。stored\_program\_definition\_cache选项为可以存储在字典对象缓存中的存储程序定义对象的数量设定了一个限制。默认值是256。

存储程序定义缓存分区与存储过程和存储函数缓存并行存在，后者是通过store\_program\_cache选项配置的。

stored\_program\_cache选项为每个连接的存储过程或函数的缓存数量设置了一个软上限，并且在每次连接执行存储过程或函数时检查这个上限。另一方面，存储程序定义缓存分区是一个共享的缓存，用于存储其他用途的存储程序定义对象。存储程序定义缓存分区中对象的存在与存储过程缓存或存储函数缓存中对象的存在没有关系，反之亦然。

字符集定义缓存分区。存储字符集定义对象，硬编码的对象限制为256。

* 整理定义缓存分区。存储整理定义对象，硬编码的对象限制为256。

关于字典对象缓存配置选项的有效值，请参阅第5.1.8节，"服务器系统变量"。

## 14.5 INFORMATION\_SCHEMA和数据字典集成

随着数据字典的引入，以下INFORMATION\_SCHEMA表被实现为数据字典表的视图。

* [**CHARACTER\_SETS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-character-sets-table)
* [**CHECK\_CONSTRAINTS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-check-constraints-table)
* [**COLLATIONS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-collations-table)
* [**COLLATION\_CHARACTER\_SET\_APPLICABILITY**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-collation-character-set-applicability-table)
* [**COLUMNS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-columns-table)
* [**COLUMN\_STATISTICS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-column-statistics-table)
* [**EVENTS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-events-table)
* [**FILES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-files-table)
* [**INNODB\_COLUMNS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-columns-table)
* [**INNODB\_DATAFILES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-datafiles-table)
* [**INNODB\_FIELDS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-fields-table)
* [**INNODB\_FOREIGN**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-foreign-table)
* [**INNODB\_FOREIGN\_COLS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-foreign-cols-table)
* [**INNODB\_INDEXES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-indexes-table)
* [**INNODB\_TABLES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-tables-table)
* [**INNODB\_TABLESPACES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-tablespaces-table)
* [**INNODB\_TABLESPACES\_BRIEF**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-tablespaces-brief-table)
* [**INNODB\_TABLESTATS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-innodb-tablestats-table)
* [**KEY\_COLUMN\_USAGE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-key-column-usage-table)
* **KEYWORDS**
* [**PARAMETERS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-parameters-table)
* [**PARTITIONS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-partitions-table)
* [**REFERENTIAL\_CONSTRAINTS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-referential-constraints-table)
* [**RESOURCE\_GROUPS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-resource-groups-table)
* [**ROUTINES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-routines-table)
* [**SCHEMATA**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-schemata-table)
* [**STATISTICS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-statistics-table)
* [**ST\_GEOMETRY\_COLUMNS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-st-geometry-columns-table)
* [**ST\_SPATIAL\_REFERENCE\_SYSTEMS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-st-spatial-reference-systems-table)
* [**TABLES**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-tables-table)
* [**TABLE\_CONSTRAINTS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-table-constraints-table)
* [**TRIGGERS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-triggers-table)
* [**VIEWS**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-views-table)
* [**VIEW\_ROUTINE\_USAGE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-view-routine-usage-table)
* [**VIEW\_TABLE\_USAGE**](file:///C:\Users\Administrator\Downloads\refman-8.0-en.html-chapter\refman-8.0-en.html-chapter\information-schema.html#information-schema-view-table-usage-table)

对这些表的查询现在更有效率了，因为他们从数据字典表获得信息，而不是通过其他更慢的方式。特别是对于每个INFORMATION\_SCHEMA表，它是数据字典表上的一个视图。

* 服务器不再必须为INFORMATION\_SCHEMA表的每次查询创建一个临时表。
* 当底层数据字典表存储了以前通过目录扫描（例如，列举数据库名称或数据库内的表名称）或文件打开操作（例如，从.frm文件中读取信息）获得的值时，INFORMATION\_SCHEMA查询这些值现在使用表查找。(另外，即使对于非视图的INFORMATION\_SCHEMA表，数据库和表名等值也是通过数据字典的查找来获取的，不需要目录或文件扫描。)
* 在底层数据字典表上的索引允许优化器构建有效的查询执行计划，这在以前的实现中是不存在的，它是通过每个查询使用一个临时表来处理 INFORMATION\_SCHEMA 表。

前面的改进也适用于显示对应于INFORMATION\_SCHEMA表的信息的SHOW语句，这些表是数据字典表的视图。例如，SHOW DATABASES显示与SCHEMATA表相同的信息。

除了在数据字典表上引入视图之外，在STATISTICS和TABLES表中包含的表的统计数据现在被缓存起来，以提高INFORMATION\_SCHEMA的查询性能。information\_schema\_stats\_expiry系统变量定义了缓存的表统计信息过期前的时间段。默认是86400秒（24小时）。如果没有缓存的统计数据或者统计数据已经过期，那么在查询表的统计列时，会从存储引擎中检索统计数据。要在任何时候更新某个表的缓存值，请使用ANALYZE TABLE

information\_schema\_stats\_expiry可以被设置为0，让INFORMATION\_SCHEMA查询直接从存储引擎检索最新的统计数据，这不如检索缓存的统计数据快。

更多信息请参见第8.2.3节，"优化INFORMATION\_SCHEMA查询"。

在MySQL 8.0中，INFORMATION\_SCHEMA表与数据字典紧密联系在一起，导致了一些使用上的差异。见第14.7节，"数据字典使用差异"。

## 14.6 串行化字典信息（SDI

除了在数据字典中存储关于数据库对象的元数据外，MySQL还以序列化的形式存储它。这种数据被称为序列化字典信息（SDI）。InnoDB在其表空间文件中存储SDI数据。NDBCLUSTER将SDI数据存储在NDB字典中。其他存储引擎将SDI数据存储在.sdi文件中，这些文件是在表的数据库目录下为一个给定的表创建的。SDI数据以一种紧凑的JSON格式生成。

串行化的字典信息（SDI）存在于所有InnoDB表空间文件中，除了临时表空间和undo tablespaces文件。InnoDB表空间文件中的SDI记录只描述表空间中包含的表和表空间对象。

SDI数据通过对表的DDL操作或CHECK TABLE FOR UPGRADE进行更新。当MySQL服务器升级到一个新版本或版本时，SDI数据不会被更新。

SDI数据的存在提供了元数据的冗余性。例如，如果数据字典变得不可用，可以使用ibd2sdi工具直接从InnoDB表空间文件中提取对象元数据。

对于InnoDB，一个SDI记录需要一个索引页，默认大小为16KB。然而，SDI数据被压缩以减少存储空间的占用。

对于由多个表空间组成的分区InnoDB表，SDI数据被存储在第一个分区的表空间文件中。

MySQL服务器使用一个内部API，在DDL操作期间访问该API以创建和维护SDI记录。

IMPORT TABLE语句根据.sdi文件中包含的信息导入MyISAM表。更多信息，请参见第13.2.5节，"IMPORT TABLE语句"。

## 14.7 数据字典使用方面的差异

与没有数据字典的服务器相比，使用启用了数据字典的MySQL服务器会带来一些操作上的差异。

* 以前，启用innodb\_read\_only系统变量可以阻止只为InnoDB存储引擎创建和删除表。从MySQL 8.0开始，启用innodb\_read\_only会阻止所有存储引擎的这些操作。任何存储引擎的表创建和删除操作都会修改mysql系统数据库中的数据字典表，但这些表使用InnoDB存储引擎，在启用innodb\_read\_only时不能被修改。这个原则也适用于其他需要修改数据字典表的表操作。例子。
  + ANALYZE TABLE失败，因为它更新了存储在数据字典中的表的统计数据。
  + ALTER TABLE tbl\_name ENGINE=engine\_name失败，因为它更新了存储引擎的名称，而这个名称是存储在数据字典中的。

注意

*启用innodb\_read\_only对mysql系统数据库中的非数据字典表也有重要影响。详情请参见第15.14节 "InnoDB启动选项和系统变量 "中对innodb\_read\_only的描述。*

* 以前，mysql系统数据库中的表对于DML和DDL语句是可见的。从MySQL 8.0开始，数据字典表是不可见的，不能被直接修改或查询。然而，在大多数情况下，有相应的INFORMATION\_SCHEMA表可以被查询。这使得底层的数据字典表可以随着服务器开发的进行而改变，同时保持一个稳定的INFORMATION\_SCHEMA接口供应用程序使用。
* 在MySQL 8.0中，INFORMATION\_SCHEMA表与数据字典紧密相连，导致了一些使用上的差异。
  + 以前，INFORMATION\_SCHEMA查询STATISTICS和TABLES表中的表的统计数据，直接从存储引擎检索统计数据。从MySQL 8.0开始，默认使用缓存的表统计数据。information\_schema\_stats\_expiry系统变量定义了缓存表统计数据过期前的时间段。默认是86400秒（24小时）。要想在任何时候更新某个表的缓存值，请使用ANALYZE TABLE）。如果没有缓存的统计信息或者统计信息已经过期，在查询表的统计列时，会从存储引擎中检索统计信息。要想总是直接从存储引擎中获取最新的统计数据，请将information\_schema\_stats\_expiry设置为0。 更多信息请参见8.2.3节 "优化INFORMATION\_SCHEMA查询"。
  + 几个INFORMATION\_SCHEMA表是数据字典表的视图，这使得优化器可以在这些底层表上使用索引。因此，根据优化器的选择，INFORMATION\_SCHEMA查询结果的行顺序可能与之前的结果不同。如果一个查询结果必须有特定的行排序特征，请包括一个ORDER BY子句。
  + 对INFORMATION\_SCHEMA表的查询可能会返回与早期MySQL系列不同的字母大写的列名。应用程序应以不区分大小写的方式测试结果集列名。如果这不可行，一个变通方法是在选择列表中使用列别名，以要求的字母大小写返回列名。比如说。

SELECT TABLE\_SCHEMA AS table\_schema, TABLE\_NAME AS table\_name

FROM INFORMATION\_SCHEMA.TABLES WHERE TABLE\_NAME = 'users';

* mysqldump和mysqlpump不再转储INFORMATION\_SCHEMA数据库，即使在命令行中明确命名。
* CREATE TABLE dst\_tbl LIKE src\_tbl要求src\_tbl是一个基表，如果它是一个INFORMATION\_SCHEMA表，是数据字典表的视图，则失败。
* 以前，从INFORMATION\_SCHEMA表中选择的列的结果集标题使用查询中指定的大写字母。这个查询产生的结果集的标题是table\_name。

SELECT table\_name FROM INFORMATION\_SCHEMA.TABLES;

从MySQL 8.0开始，这些头是大写的；前面的查询产生一个头为TABLE\_NAME的结果集。如果有必要，可以用一个列的别名来实现不同的字母大小写。例如。

SELECT table\_name AS 'table\_name' FROM INFORMATION\_SCHEMA.TABLES;

* 数据目录会影响mysqldump和mysqlpump从mysql系统数据库转储信息的方式。
* 以前，可以转储mysql系统数据库中的所有表。从MySQL 8.0开始，mysqldump和mysqlpump只转储该数据库中的非数据字典表。
  + 以前，可以转储mysql系统数据库中的所有表。从MySQL 8.0开始，mysqldump和mysqlpump只转储该数据库中的非数据字典表。
  + 以前，当使用--all-databases选项时，不需要--routines和--events选项来包括存储例程和事件。转储包括mysql系统数据库，因此也包括含有存储例程和事件定义的proc和事件表。从MySQL 8.0开始，事件表和proc表不被使用。相应对象的定义被存储在数据字典表中，但这些表没有被转储。要在使用--all-databases进行的转储中包括存储的例程和事件，请明确使用--routines和--events选项。
  + 以前， --routines选项需要proc表的SELECT权限。从MySQL 8.0开始，该表不被使用；--routines需要全局SELECT权限来代替。
  + 以前，可以通过转储proc和event表来转储存储的例程和事件定义，以及它们的创建和修改时间戳。从MySQL 8.0开始，这些表不被使用，所以不可能转储时间戳。
* 以前，创建一个包含非法字符的存储例程会产生一个警告。从MySQL 8.0开始，这就是一个错误。

## 14.8 数据字典的限制

本节介绍了用MySQL数据字典引入的临时限制。

* 不支持在数据目录下手动创建数据库目录（例如，用mkdir）。手动创建的数据库目录不被MySQL服务器所识别。
* 由于向存储、Undo logs和redo log而不是.frm文件写入，DDL操作需要更长的时间。