Changed in v2.0: 查询可以指定表名，而不需要数据库或schema名（例如，SELECT \* FROM orders）。 CockroachDB如何知道使用的是哪个orders表以及在哪个schema下？

本文详细介绍了CockroachDB如何执行**名称解析**以回答此问题。

逻辑schema和命名空间

New in v2.0: 逻辑schema和CockroachDB集群可以存储多个数据库，每个数据库可以存储多个表/视图/序列。 这种用于存储数据的**两级结构**在关系数据库管理系统中通常称为“逻辑schema”。

同时，CockroachDB旨在提供与PostgreSQL客户端应用程序的兼容性，从而支持PostgreSQL的SQL查询语义。 为实现这一目标，CockroachDB支持名称的三级结构。 这称为“分层命名”。

在命名分层结构中，存储对象的路径有三个组件：

* 数据库名 (也称为"catalog")
* schema名
* 对象名

任何给定数据库中所有存储对象的schema名称始终是public的。 存储对象只有一个可用的schema，因为CockroachDB仅支持两级存储结构。

除了public之外，CockroachDB还支持一组固定的虚拟schema，可在每个数据库中使用，为客户端应用程序提供辅助功能，不用作数据存储。 例如，提供information\_schema是为了与SQL标准兼容。

可以使用SHOW DATABASES获取所有数据库的列表。 可以使用SHOW SCHEMAS获取给定数据库的所有schema的列表。 可以使用其他SHOW语句获取给定schema的所有对象的列表。

名称解析如何工作?

名字解析分别发生在“查找现有对象“和“定义新对象全名“。

查找现有对象的规则如下：

1. 如果名称已完全指定数据库和schema，请使用该信息。
2. 如果名称具有单个组件前缀，尝试在当前数据库中查找具有前缀名称的schema。 如果失败，尝试使用前缀名称查找在数据库的public schema中的对象。
3. 如果名称没有前缀，请使用当前数据库的搜索路径。

类似的，定义新对象全名的规则如下：

1. 如果名称已完全指定数据库和schema，使用该名称。
2. 如果名称具有单个组件前缀，尝试查找具有该名称的schema。 如果不存在此类schema，请使用数据库中的带有前缀名称的public schema。
3. 如果名称没有前缀，请使用当前数据库中的当前schema。

名称解析的参数

当前数据库

当名称不合格或只有一个组件前缀时，将使用当前数据库。 它是database session变量的当前值。

* 可以使用SHOW database查看database会话变量的当前值，并使用SET database进行更改
* 你可以查看有效数据库的列表，可以通过SHOW database来查看指定数据库
* 对于使用postgres://...形式的URL连接到CockroachDB的客户端应用程序，可以使用URL的路径组件设置database会话变量的初始值。 例如，postgres//node/mydb在建立连接时将database设置为mydb。

查找路径

当名称不合格（没有前缀）时使用搜索路径。 它列出了查找对象的schema。 它的第一个元素也是创建新对象的当前schema。

* 你可以使用SET search\_path设置当前搜索路径，并使用SHOW search\_path进行检查
* 你可以使用SHOW SCHEMAS检查可以在search\_path中列出的有效schema列表
* 默认情况下，搜索路径包含public和pg\_catalog。 为了与PostgreSQL兼容，pg\_catalog被强制一直存在于search\_path中，即使没有用SET search\_path指定。

当前Schema

如果名称不合格（没有前缀），则在创建新对象时将当前schema用作目标schema。

* 为了与PostgreSQL兼容，当前schema始终是search\_path的第一个值。
* 你可以使用特殊的内置函数/标识符current\_schema()检查当前schema。

示例

以下示例使用逻辑schema作为起点：

> **CREATE** **DATABASE** mydb;

> **CREATE** **TABLE** mydb.mytable(x INT);

> **SET** **database** = mydb;

使用不合格名称查找

一个不合格名称是没有前缀的名称，即简单标识符。

> **SELECT** \* **FROM** mytable;

这使用当前数据库上的搜索路径。 默认情况下，搜索路径为public，位于当前数据库中。 已解析的名称是mydb.public.mytable。

> **SET** **database** = system;

> **SELECT** \* **FROM** mytable;

pq: relation "mytable" **does** **not** exist

这使用当前数据库的搜索路径，目前是system。 搜索路径中没有schema包含表mytable，因此查找失败并显示错误。

使用完全合格的名称查找

完全合格的名称是指具有两个前缀组件的名称，即由句点分隔的三个标识符。

> **SELECT** \* **FROM** mydb.**public**.mytable;

数据库和schema组件都指定了。 当且仅当对象存在于该特定位置时，查找才会成功。

使用部分合格的名称查找

部分合格的名称是指具有一个前缀组件的名称，即由句点分隔的两个标识符。 当名称部分合格时，CockroachDB将首先尝试使用前缀作为schema名称; 如果失败，请将其用作数据库名称。

举个例子:

> **SELECT** \* **FROM** **public**.mytable;

这会在当前数据库的public schema中查找mytable。 如果当前数据库是mydb，则查找成功。

为了与CockroachDB 1.x兼容，并简化多数据库场景中的开发，CockroachDB还允许查询以部分合格名称指定数据库名称。

举个例子:

> **SELECT** \* **FROM** mydb.mytable;

在这种情况下，CockroachDB将首先尝试在当前数据库中找到一个名为mydb的schema。 当不存在这样的schema时（本节中开始的例子就是这种情况），它会尝试找到一个名为mydb的数据库并在其中使用public schema。 在此示例中，此规则适用，完全解析的名称为mydb.public.mytable。

使用搜索路径跨schema使用表

假设客户端经常访问Information Schema中存储的表以及虚拟表，由于information\_schema默认情况下不在搜索路径中，因此需要访问它的所有查询都必须明确指定。

举个例子:

> **SELECT** \* **FROM** mydb.information\_schema.schemata; *-- valid*

> **SELECT** \* **FROM** information\_schema.schemata; *-- valid; uses mydb implicitly*

> **SELECT** \* **FROM** schemata; *-- invalid; information\_schema not in search\_path*

对于经常使用information\_schema的客户端，你可以将其添加到搜索路径以简化查询. 举个例子:

> **SET** search\_path = **public**, information\_schema;

> **SELECT** \* **FROM** schemata; *-- now valid, uses search\_path*

具有特殊名称的数据库

当仅使用一个组件前缀解析部分合格名称时，CockroachDB将首先查找具有给定前缀名称的schema，并且只有在schema查找失败时才查找具有该名称的数据库。 这在（可能不常见的）希望你的数据库名为information\_schema，public，pg\_catalog或crdb\_internal的情况下较重要。

举个例子:

> **CREATE** **DATABASE** **public**;

> **SET** **database** = mydb;

> **CREATE** **TABLE** **public**.mypublictable (x INT);

此示例中的CREATE TABLE语句使用部分合格名称。 因为public前缀在当前数据库中指定了有效的schema，所以mypublictable的全名变为mydb.public.mypublictable。 该表在数据库mydb中创建。

要在数据库public中创建表，可以改为使用完全合格名称，如下所示：

> **CREATE** **DATABASE** **public**;

> **CREATE** **TABLE** **public**.**public**.mypublictable (x INT);

See Also

* [SET](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SET(session-variable)/)
* [SHOW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW(session-variables)/)
* [SHOW DATABASES](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-DATABASES/)
* [SHOW SCHEMAS](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-SCHEMAS/)
* [Information Schema](http://doc.cockroachchina.baidu.com/#develop/information-schema/)