视图是选择查询的存储，并为其提供简写名称。CockroachDB的视图是非物质化的：它不存储查询的结果，而是在每次使用视图时都重新执行查询。

为什么使用视图？

有几种情况会使用视图：

* 隐藏查询的复杂性
* 限制对底层数据的访问

隐藏查询的复杂性

当你使用一个复杂查询，例如join几个table或者执行复杂的计算，你可以将查询存储为view，然后从这个view上进行select，就像在一个普通的table上执行一样。

示例

假设你现在有一个startrek数据库，它包含两个tables：episodes和quotes。episodes.id列和quotes.episode 列有外键约束。为了计算每个人的名言数量，你可以使用以下join语句：

> **SELECT** startrek.episodes.season, **count**(\*)

**FROM** startrek.quotes

**JOIN** startrek.episodes

**ON** startrek.quotes.episode = startrek.episodes.id

**GROUP** **BY** startrek.episodes.season;

+--------+----------+

| season | count(\*) |

+--------+----------+

| 2 | 76 |

| 3 | 46 |

| 1 | 78 |

+--------+----------+

(3 rows)

此外, 为了让这个复杂语句更简单，可以创建一个视图：

> **CREATE** **VIEW** startrek.quotes\_per\_season (season, quotes)

**AS** **SELECT** startrek.episodes.season, **count**(\*)

**FROM** startrek.quotes

**JOIN** startrek.episodes

**ON** startrek.quotes.episode = startrek.episodes.id

**GROUP** **BY** startrek.episodes.season;

**CREATE** **VIEW**

然后，在这个视图上执行select语句：

> **SELECT** \* **FROM** startrek.quotes\_per\_season;

+--------+--------+

| season | quotes |

+--------+--------+

| 2 | 76 |

| 3 | 46 |

| 1 | 78 |

+--------+--------+

(3 rows)

限制对底层数据的访问

当你不想对某个用户授予对一个或多个表的所有数据的访问权限时，你可以创造一个包含特定列/行的视图，然后授权给用户访问。

示例

假设你有一个bank数据库，包含一个accounts表：

> **SELECT** \* **FROM** bank.accounts;

+----+----------+---------+-----------------+

| id | type | balance | email |

+----+----------+---------+-----------------+

| 1 | checking | 1000 | max@roach.com |

| 2 | savings | 10000 | max@roach.com |

| 3 | checking | 15000 | betsy@roach.com |

| 4 | checking | 5000 | lilly@roach.com |

| 5 | savings | 50000 | ben@roach.com |

+----+----------+---------+-----------------+

(5 rows)

你想让特定用户bob可以在不查看每个帐户余额的情况下查看每个用户拥有的帐户类型，你可以创建一个视图只包含type和email列：

> **CREATE** **VIEW** bank.user\_accounts

**AS** **SELECT** type, email

**FROM** bank.accounts;

**CREATE** **VIEW**

然后确认下bob用户没有访问bank.accounts表的权限：

> **SHOW** **GRANTS** **ON** bank.accounts;

+----------+------+------------+

| Table | User | Privileges |

+----------+------+------------+

| accounts | root | ALL |

| accounts | toti | SELECT |

+----------+------+------------+

(2 rows)

最后，为用户bob授予bank.user\_accounts视图的权限：

> **GRANT** **SELECT** **ON** bank.user\_accounts **TO** bob;

现在，bob用户如果尝试访问bank.accounts表将会报错，不过它可以正常访问 bank.user\_accounts视图。

> **SELECT** \* **FROM** bank.accounts;

pq: user bob **does** **not** have SELECT privilege **on** table accounts

> **SELECT** \* **FROM** bank.user\_accounts;

+----------+-----------------+

| type | email |

+----------+-----------------+

| checking | max@roach.com |

| savings | max@roach.com |

| checking | betsy@roach.com |

| checking | lilly@roach.com |

| savings | ben@roach.com |

+----------+-----------------+

(5 rows)

视图是如何工作的？

创建视图

使用CREATE VIEW创建一个视图：

> **CREATE** **VIEW** bank.user\_accounts

**AS** **SELECT** type, email

**FROM** bank.accounts;

**CREATE** **VIEW**

任何选择查询都可以作为CREATE VIEW的有效操作数，而不仅仅是简单的select子句。

列出视图

视图创建后，将与数据库中的常规表一起列出：

> **SHOW** **TABLES** **FROM** bank;

+---------------+

| Table |

+---------------+

| accounts |

| user\_accounts |

+---------------+

(2 rows)

如果只想看视图部分，你可以在Information Schema中查询views 表：

> **SELECT** \* **FROM** bank.information\_schema.views;

> **SELECT** \* **FROM** startrek.information\_schema.views;

+---------------+-------------------+----------------------+---------------------------------------------+--------------+--------------+--------------------+----------------------+----------------------+----------------------------+

| table\_catalog | table\_schema | table\_name | view\_definition | check\_option | is\_updatable | is\_insertable\_into | is\_trigger\_updatable | is\_trigger\_deletable | is\_trigger\_insertable\_into |

+---------------+-------------------+----------------------+---------------------------------------------+--------------+--------------+--------------------+----------------------+----------------------+----------------------------+

| bank | public | user\_accounts | SELECT type, email FROM bank.accounts | NULL | NULL | NULL | NULL | NULL | NULL |

+---------------+-------------------+----------------------+---------------------------------------------+--------------+--------------+--------------------+----------------------+----------------------+----------------------------+

(1 row)

+---------------+-------------------+----------------------+---------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------------+--------------+--------------------+----------------------+----------------------+----------------------------+

| table\_catalog | table\_schema | table\_name | view\_definition | check\_option | is\_updatable | is\_insertable\_into | is\_trigger\_updatable | is\_trigger\_deletable | is\_trigger\_insertable\_into |

+---------------+-------------------+----------------------+---------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------------+--------------+--------------------+----------------------+----------------------+----------------------------+

| startrek | public | quotes\_per\_season | SELECT startrek.episodes.season, count(\*) FROM startrek.quotes JOIN startrek.episodes ON startrek.quotes.episode = startrek.episodes.id GROUP BY startrek.episodes.season | NULL | NULL | NULL | NULL | NULL | NULL |

+---------------+-------------------+----------------------+---------------------------------------------------------------------------------------------------------------------------------------------------------------------------+--------------+--------------+--------------------+----------------------+----------------------+----------------------------+

(1 row)

查询视图

要查询视图，请使用table表达式对其进行定位，例如使用SELECT子句，就像使用存储表一样：

> **SELECT** \* **FROM** bank.user\_accounts;

+----------+-----------------+

| type | email |

+----------+-----------------+

| checking | max@roach.com |

| savings | max@roach.com |

| checking | betsy@roach.com |

| checking | lilly@roach.com |

| savings | ben@roach.com |

+----------+-----------------+

(5 rows)

选择视图会执行视图存储的SELECT语句，该语句返回基础表中的相关数据。 要检查视图执行的SELECT语句，请使用SHOW CREATE VIEW语句：

> **SHOW** **CREATE** **VIEW** bank.user\_accounts;

+--------------------+---------------------------------------------------------------------------+

| View | CreateView |

+--------------------+---------------------------------------------------------------------------+

| bank.user\_accounts | CREATE VIEW "bank.user\_accounts" AS SELECT type, email FROM bank.accounts |

+--------------------+---------------------------------------------------------------------------+

(1 row)

您还可以通过查询Information Schema中的views表来检查视图执行的SELECT语句：

> **SELECT** view\_definition **FROM** bank.information\_schema.views **WHERE** table\_name = 'user\_accounts';

+----------------------------------------+

| view\_definition |

+----------------------------------------+

| SELECT type, email FROM bank.accounts |

+----------------------------------------+

(1 row)

视图依赖

视图取决于其基础查询所针对的对象。 因此，尝试重命名视图的存储查询中引用的对象会导致错误：

> **ALTER** **TABLE** bank.accounts **RENAME** **TO** bank.accts;

pq: cannot **rename** **table** "bank.accounts" because **view** "user\_accounts" depends **on** it

同样，尝试删除视图的存储查询中引用的对象会导致错误：

> **DROP** **TABLE** bank.accounts;

pq: cannot **drop** **table** "accounts" because **view** "user\_accounts" depends **on** it

> **ALTER** **TABLE** bank.accounts **DROP** **COLUMN** email;

pq: cannot **drop** **column** email because **view** "bank.user\_accounts" depends **on** it

但是，上面的规则有一个例外：当删除表或删除视图时，你也可以使用CASCADE关键字删除所有依赖对象：

> **DROP** **TABLE** bank.accounts **CASCADE**;

**DROP** **TABLE**

注意：CASCADE删除所有相关对象时并不列出它们，这可能导致无意和难以恢复的损失。 为避免潜在的伤害，我们建议在大多数情况下单独删除对象。

视图重命名

可以使用ALTER VIEW语句重命名视图：

> **ALTER** **VIEW** bank.user\_accounts **RENAME** **TO** bank.user\_accts;

**RENAME** **VIEW**

无法更改视图执行的存储查询，必须删除现有视图并创建新视图。

删除视图

要删除视图，请使用DROP VIEW语句：

> **DROP** **VIEW** bank.user\_accounts

**DROP** **VIEW**

See Also

* [Selection Queries](http://doc.cockroachchina.baidu.com/#develop/sql-syntax/selection-queries/)
* [Simple SELECT Clauses](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SELECT/)
* [CREATE VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/CREATE-VIEW/)
* [SHOW CREATE VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/SHOW-CREATE-VIEW/)
* [GRANT](http://doc.cockroachchina.baidu.com/#develop/sql-statements/GRANT-%3Cprivileges%3E/)
* [ALTER VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALERT-VIEW/)
* [DROP VIEW](http://doc.cockroachchina.baidu.com/#develop/sql-statements/DROP-VIEW/)