（New in v2.0） CockroachDB允许用户定义分区，能够获得行级别的控制数据存储位置的能力，通过分区功能也可以达到减少延迟、节约成本的效果，此外帮助用户满足在数据存储方面符合地方法律规定的需求。

**NOTE:** 该功能为企业版功能，需要[企业版License](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/)

使用场景

分区能够减少延迟，节约成本：

* 地理分区 允许用户将用户数据存储在用户附近的节点，从而减少数据传输的距离，因此能够**减少延迟**。若用户要对表进行地理分区，需要在创建表时定义基于地理位置的分区规则，创建特定地理位置的区间配置，并将区间配置应用于相应的地理分区。
* 归档分区 允许用户在较慢和较便宜的存储上存储不被经常访问到的数据，达到**节约成本**的目的。若用户要对表进行归档分区，需要在创建表时定义基于访问频率的分区规则，创建基于访问频率指定数据存储到适当存储设备的区间配置，并将区间配置应用到相应的归档分区。

工作原理

分区功能通过组合以下功能模块实现。

节点属性

配置地理分区或是规范分区，需要在[节点启动](http://doc.cockroachchina.baidu.com/#deploy/start-a-node/)的时候指定以下Flag：

* 使用--locality指定节点位置相关的键值对序列，例如--locality=region=east,datacenter=us-east-1。
* 使用--attrs指定节点硬件条件，包含CPU数量或是配置的硬件类型，例如--attrs=ram:64gb。
* 使用--store的attrs属性指定硬盘类型和大小，例如--store=path=/mnt/ssd01,attrs=ssd。

企业版License

使用分区功能需要用户具备企业版License，关于获取并应用试用或是正式的企业版License，可查看[企业版License](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/)。

对于过期的企业版License，以下功能能够继续使用：

* 查询分区，例如SELECT foo PARTITION。
* 向分区表插入和更新数据
* 删除分区
* 取消分区
* 对分区表使用分区以外的其他功能，例如变更表结构、添加索引、添加约束等。

同时，用户不能再使用以下功能：

* 创建新的分区
* 为分区添加新的区间配置
* 变更表或索引的分区规则
* 变更指定分区的区间配置

创建表

用户可以基于表其中一列或多列属性定义分区和子分区。在[创建表](https://www.cockroachlabs.com/docs/stable/create-table.html)期间，用户可以通过以下两种方式之一定义每个分区的值：

* **基于列表分区**：列出每个分区的所有可能值，适用于可能值数量不多的情况，例如地理分区。
* **基于Range分区**：指定每个分区指定一个连续的值范围的下限和上限，适用于可能值过多或不可枚举的情况，例如归档分区。

基于列表分区

用户可以使用[PARTITION BY LIST](https://www.cockroachlabs.com/docs/stable/partition-by.html)语句在创建表的时候定义分区规则，对于不满足所有用户自定义分区规则的数据将映射到DEFAULT分区，详见[Partition by List](http://doc.cockroachchina.baidu.com/#deploy/define-table-partitions-(enterprise)/#partition-by-list)。

基于范围分区

用户可以使用[PARTITION BY RANGE](https://www.cockroachlabs.com/docs/stable/partition-by.html)语句在创建表的时候定义分区规则。使用该语句需要指定MINVALUE和MAXVALUE参数，分别对应指定分区的可能值的最大值和最小值。

**NOTE:** 定义范围区间时包含最小值，不包含最大值。 与KEY编码排序和ORDER BY语句的逻辑一致，认为NULL值小于所有可能值。

分区范围的值可以使用SQL查询的查询结果值，但首次赋值后不再执行该SQL查询。例如用户在2017年1月30日使用< (now() - '1d')定义分区，对应着所有不晚于2017年1月29日的值，不会随着时间而改变分区范围。

基于主键建立分区

表分区使用的主键跟一般的主键有所不同。在定义表分区的联合主键的时候，需要将分区和子分区关联的主键提前，且分区主键前于子分区主键。 例如，考虑一个全球在线学习门户的数据库，该门户网站为世界各地所有课程的学生创建一张表。如果要根据学生所在国家对表数据进行地理分区，则需要将主键定义为：

**CREATE** **TABLE** students (

id SERIAL,

name **STRING**,

email **STRING**,

country **STRING**,

expected\_graduation\_date DATE,

**PRIMARY** **KEY** (country, id));

用户须知

* 对于v2.0版本，用户无法在创建表以后修改主键，因此用户需要在创建表的时候提前将以后需要用于分区的属性列加入主键当中。在上述全球在线学习门户的例子，如果用户需要基于expected\_graduation\_date做子分区，则需要提前将主键定义为(country, expected\_graduation\_date, id)。

**NOTE:** v2.1版本后则允许修改主键。

* 主键定义顺序十分重要，分区主键需要在子分区主键前面。在上述全球在线学习门户的例子中定义主键为(country, expected\_graduation\_date, id)，则CockroachDB底层先基于country属性做分区，再基于expected\_graduation\_date属性做子分区。值得注意的是，这种情况下无法跳过country属性，直接基于expected\_graduation\_date属性做分区。

基于二级索引建立分区

上述基于主键的分区有两个缺点：

* 无法保证用于分区的属性列组合是全局唯一的。
* 无法对属性列组合进行快速查找

解决以上问题的方式创建一个能够快速查找的唯一索引，并基于该二级索引建立分区。

值得注意的是，基于主键和基于该表的二级索引的所有分区的分区名不能重复。例如，以下分区定义的过程中，CREATE INDEX语句会因为重复分区名bar而执行失败：

**CREATE** **TABLE** foo (a **STRING** **PRIMARY** **KEY**, b **STRING**) **PARTITION** **BY** LIST (a) (

bar **VALUES** **IN** ('bar'),

**default** **VALUES** **IN** (**DEFAULT**)

);

**CREATE** **INDEX** foo\_b\_idx **ON** foo (b) **PARTITION** **BY** LIST (b) (

baz **VALUES** **IN** ('baz'),

**default** **VALUES** **IN** (**DEFAULT**)

);

用户需要定义一套命名规则避免命名重复，例如命名为primary\_idx\_bar、primary\_idx\_default、b\_idx\_baz和b\_idx\_default。

对交错表进行表分区

对于[交错表](https://www.cockroachlabs.com/docs/stable/interleave-in-parent.html)的分区功能，目前只允许在父表定义分区，而子表与父表保持一致。

复制区域

对于满足分区规则的行将插入到对应的分区当中。用户可以针对分区创建和应用相应的[复制区域](http://doc.cockroachchina.baidu.com/#deploy/configure-replication-zones/)。目前分区相较于表、索引或是数据库，是复制空间的最小粒度的配置单位。

示例

基于列表定义表分区

假设搭建一个全球在线学习门户网站RoachLearn，其数据库包含一张存储世界各地学生数据的表。用户将数据库部署在两个数据中心，一个在美国，另一个在澳大利亚。为了减少延迟，用户希望让学生的数据存储在地理上更接近他们的数据中心当中：

* 需要将美国和加拿大学生的数据存储在美国的数据中心
* 需要将澳大利亚和新西兰学生的数据存储在澳大利亚的数据中心

Step 1: 确定分区方式

此场景下，用户需要使用PARTITION BY LIST语句，基于country属性对学生表进行分区。

Step 2: 启动节点

需要在节点启动的时候使用--localityFlag指定节点对应的数据中心。

*# Start the node in the US datacenter:*

cockroach start --insecure --locality=datacenter=us1 --store=node1 --host=<node1 hostname> --port=26257 --http-port=8080 --join=<node1 hostname>:26257,<node2 hostname>:26258

*# Start the node in the AUS datacenter:*

cockroach start --insecure --locality=datacenter=au1 --store=node2 --host=<node2 hostname> --port=26258 --http-port=8081 --join=<node1 hostname>:26257,<node2 hostname>:26258

Step 3: 配置企业版License

详见[配置企业版License](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/#set-the-trial-or-enterprise-license-key)

Step 4: 创建表

用户需要在创建表的同时定义分区：

**CREATE** **TABLE** students\_by\_list (

id SERIAL,

name **STRING**,

email **STRING**,

country **STRING**,

expected\_graduation\_date DATE,

**PRIMARY** **KEY** (country, id))

**PARTITION** **BY** LIST (country)

(**PARTITION** north\_america **VALUES** **IN** ('CA','US'),

**PARTITION** australia **VALUES** **IN** ('AU','NZ'),

**PARTITION** **DEFAULT** **VALUES** **IN** (**default**));

Step 5: 创建并应用区间配置

* 为数据中心创建区间配置文件

cat > north\_america.zone.yml

constraints: [+datacenter=us1]

cat > australia.zone.yml

constraints: [+datacenter=au1]

* 将配置文件应用到相应分区

cockroach zone set roachlearn.students\_by\_list.north\_america --insecure -f north\_america.zone.yml

cockroach zone set roachlearn.students\_by\_list.australia --insecure -f australia.zone.yml

Step 6: 验证表分区

SHOW EXPERIMENTAL\_RANGES FROM TABLE students\_by\_list;

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

| Start Key | End Key | Range ID | Replicas | Lease Holder |

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

| NULL | /"AU" | 251 | {1,2,3} | 1 |

| /"AU" | /"AU"/PrefixEnd | 257 | {1,2,3} | 1 |

| /"AU"/PrefixEnd | /"CA" | 258 | {1,2,3} | 1 |

| /"CA" | /"CA"/PrefixEnd | 252 | {1,2,3} | 1 |

| /"CA"/PrefixEnd | /"NZ" | 253 | {1,2,3} | 1 |

| /"NZ" | /"NZ"/PrefixEnd | 256 | {1,2,3} | 1 |

| /"NZ"/PrefixEnd | /"US" | 259 | {1,2,3} | 1 |

| /"US" | /"US"/PrefixEnd | 254 | {1,2,3} | 1 |

| /"US"/PrefixEnd | NULL | 255 | {1,2,3} | 1 |

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

(9 rows)

Time: 7.209032ms

基于范围定义表分区

该场景中，假定用户需要将当期学生的数据存储在快速且昂贵的存储设备（例如SSD）上，而将毕业学生的数据存储在较慢且较便宜的存储设备（例如HDD）上。

Step 1: 确定分区方式

用户需要使用PARTITION BY RANGE语句基于时间进行表分区。

Step 2: 配置企业版License

详见[配置企业版License](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/#set-the-trial-or-enterprise-license-key)

Step 3: 启动节点

需要在节点启动的时候使用--storeFlag指定节点对应的存储设备。

cockroach start --insecure --store=path=/mnt/1,attrs=ssd --host=<node1 hostname> --port=26257 --http-port=8080 --join=<node1 hostname>:26257,<node2 hostname>:26258

cockroach start --insecure --store=path=/mnt/2,attrs=hdd --host=<node2 hostname> --port=26258 --http-port=8081 --join=<node1 hostname>:26257,<node2 hostname>:26258

Step 4: 创建表

用户需要在创建表的同时定义分区：

**CREATE** **TABLE** students\_by\_range (

id SERIAL,

name **STRING**,

email **STRING**,

country **STRING**,

expected\_graduation\_date DATE,

**PRIMARY** **KEY** (expected\_graduation\_date, id))

**PARTITION** **BY** RANGE (expected\_graduation\_date)

(**PARTITION** graduated **VALUES** **FROM** (MINVALUE) **TO** ('2017-08-15'),

**PARTITION** **current** **VALUES** **FROM** ('2017-08-15') **TO** (MAXVALUE));

Step 5: 创建并应用区间配置

* 为不同类型的存储设备创建区间配置文件

cat > current.zone.yml

constraints: [+ssd]

cat > graduated.zone.yml

constraints: [+hdd]

* 将配置文件应用到相应分区

cockroach zone **set** roachlearn.students\_by\_range.**current** *--insecure -f current.zone.yml*

cockroach zone **set** roachlearn.students\_by\_range.graduated *--insecure -f graduated.zone.yml*

Step 6: 验证表分区

**SHOW** EXPERIMENTAL\_RANGES **FROM** **TABLE** students\_by\_range;

+*-----------+---------+----------+----------+--------------+*

| **Start** **Key** | **End** **Key** | Range ID | Replicas | Lease Holder |

+*-----------+---------+----------+----------+--------------+*

| NULL | /17393 | 244 | {1,2,3} | 1 |

| /17393 | NULL | 242 | {1,2,3} | 1 |

+*-----------+---------+----------+----------+--------------+*

(2 **rows**)

**Time**: 5.850903ms

定义表的子分区

分区能被再分区，形成子分区，子分区的层次数量没有限制。

**NOTE:** 基于范围的分区也能创建子分区

在RoachLearn的场景中，假定用户需要实现：

* 将学生的数据存储在地理上更接近他们的数据中心
* 将当期学生的数据存储在快速存储设备上
* 将毕业学生的数据存储在慢速且廉价的存储设备上

Step 1: 确定分区方式

用户需要先进行地理分区，在依据时间进行子分区。

Step 2: 启动节点

使用--storeFlag指定节点使用的存储设备

* 启动部署在美国数据中心的节点

cockroach start --insecure --host=<node1 hostname> --locality=datacenter=us1 --store=path=/mnt/1,attrs=ssd --store=path=/mnt/2,attrs=hdd \

* 启动部署在澳大利亚数据中心的节点

cockroach start --insecure --host=<node2 hostname> --locality=datacenter=au1 --store=path=/mnt/3,attrs=ssd --store=path=/mnt/4,attrs=hdd --join=<node1 hostname>:26257

* 初始化集群

*cockroach* *init* --*insecure* --*host=*<*node1* *hostname*>

Step 3: 配置企业版License

详见[配置企业版License](http://doc.cockroachchina.baidu.com/#deploy/enterprise-licensing/#set-the-trial-or-enterprise-license-key)

Step 4: 创建表

用户需要在创建表的同时定义分区：

**CREATE** **TABLE** students (

id SERIAL,

name **STRING**,

email **STRING**,

country **STRING**,

expected\_graduation\_date DATE,

**PRIMARY** **KEY** (country, expected\_graduation\_date, id))

**PARTITION** **BY** LIST (country)(

**PARTITION** australia **VALUES** **IN** ('AU','NZ') **PARTITION** **BY** RANGE (expected\_graduation\_date)(**PARTITION** graduated\_au **VALUES** **FROM** (MINVALUE) **TO** ('2017-08-15'), **PARTITION** current\_au **VALUES** **FROM** ('2017-08-15') **TO** (MAXVALUE)),

**PARTITION** north\_america **VALUES** **IN** ('US','CA') **PARTITION** **BY** RANGE (expected\_graduation\_date)(**PARTITION** graduated\_us **VALUES** **FROM** (MINVALUE) **TO** ('2017-08-15'), **PARTITION** current\_us **VALUES** **FROM** ('2017-08-15') **TO** (MAXVALUE))

);

分区和子分区的分区名不允许重复，因此推荐用户定义一套命名规则命名分区，例如使用graduated\_au、graduated\_us、and current\_au和current\_us。

Step 5: 创建并应用区间配置

* 创建多个区间配置文件

cat > current\_us.zone.yml

constraints: [+ssd,+datacenter=us1]

cat > graduated\_us.zone.yml

constraints: [+hdd,+datacenter=us1]

cat > current\_au.zone.yml

constraints: [+ssd,+datacenter=au1]

cat > graduated\_au.zone.yml

constraints: [+hdd,+datacenter=au1]

* 将配置文件应用到相应分区

cockroach zone set roachlearn.students.current\_us --insecure -f current\_us.zone.yml

cockroach zone set roachlearn.students.graduated\_us --insecure -f graduated\_us.zone.yml

cockroach zone set roachlearn.students.current\_au --insecure -f current\_au.zone.yml

cockroach zone set roachlearn.students.graduated\_au --insecure -f graduated\_au.zone.yml

Step 6: 验证表分区

**SHOW** EXPERIMENTAL\_RANGES **FROM** **TABLE** students;

+*-----------------+-----------------+----------+----------+--------------+*

| **Start** **Key** | **End** **Key** | Range ID | Replicas | Lease Holder |

+*-----------------+-----------------+----------+----------+--------------+*

| NULL | /"AU" | 260 | {1,2,3} | 1 |

| /"AU" | /"AU"/17393 | 268 | {1,2,3} | 1 |

| /"AU"/17393 | /"AU"/PrefixEnd | 266 | {1,2,3} | 1 |

| /"AU"/PrefixEnd | /"CA" | 267 | {1,2,3} | 1 |

| /"CA" | /"CA"/17393 | 265 | {1,2,3} | 1 |

| /"CA"/17393 | /"CA"/PrefixEnd | 261 | {1,2,3} | 1 |

| /"CA"/PrefixEnd | /"NZ" | 262 | {1,2,3} | 3 |

| /"NZ" | /"NZ"/17393 | 284 | {1,2,3} | 3 |

| /"NZ"/17393 | /"NZ"/PrefixEnd | 282 | {1,2,3} | 3 |

| /"NZ"/PrefixEnd | /"US" | 283 | {1,2,3} | 3 |

| /"US" | /"US"/17393 | 281 | {1,2,3} | 3 |

| /"US"/17393 | /"US"/PrefixEnd | 263 | {1,2,3} | 1 |

| /"US"/PrefixEnd | NULL | 264 | {1,2,3} | 1 |

+*-----------------+-----------------+----------+----------+--------------+*

(13 **rows**)

**Time**: 11.586626ms

修改表分区

在RoachLearn场景中，假定学生表已经基于时间范围进行分区，将当期学生存储在快速且昂贵的存储设备（例如：SSD）上，将毕业学生的数据存储在速度较慢且低廉的存储设备上（例如：HDD）。现在用户需要变更时间范围，定义当期学生为2018-08-15之后，则需要使用[ALTER TABLE](http://doc.cockroachchina.baidu.com/#develop/sql-statements/ALTER-TABLE/)和[PARTITION BY](https://www.cockroachlabs.com/docs/stable/partition-by.html)语句修改分区规则。

**ALTER** **TABLE** students\_by\_range **PARTITION** **BY** RANGE (expected\_graduation\_date) (

**PARTITION** graduated **VALUES** **FROM** (MINVALUE) **TO** ('2018-08-15'),

**PARTITION** **current** **VALUES** **FROM** ('2018-08-15') **TO** (MAXVALUE));

取消分区

使用[PARTITION BY NOTHING](https://www.cockroachlabs.com/docs/stable/partition-by.html)取消指定表的分区。

**ALTER** **TABLE** students **PARTITION** **BY** NOTHING;

性能与可靠性之间的权衡

在提高读/写性能和集群可靠性上存在权衡，假设澳大利亚学生的数据在数据冗余上配置了3份副本。

* 如果位于澳大利亚数据中心的节点只有一份数据副本，由于**follow-the-workload**机制导致Range租约在该唯一副本上，此时在性能上是读快写慢。
* 如果位于澳大利亚数据中心的节点有2份数据副本，此时在性能上读写都很快。但考虑到如果两份数据副本在同一数据中心，该数据中心若出现故障，则会导致集群数据不可用。因此为了可靠性，用户可能需要将2个节点分别部署在澳大利亚两个不同的数据中心。
* 如果位于澳大利亚数据中心的节点有3份数据副本，此时推荐将3个节点分别部署在澳大利亚3个不同的数据中心。

对比其他数据库

其他数据库的分区功能相较于CockroachDB，拥有3个额外的使用场景：

* **二级索引变更**：CockroachDB能够通过在线schema变更功能解决索引变更问题。在线schema变更功能是分区功能的高级特性，能够在不停机的情况下变更schema，同时保障数据的一致性。
* **数据拆分**：CockroachDB能够自动拆分数据到分布式数据库当中。
* **批量导入和删除**：CockroachDB目前不支持。