

Expiring data with time-to-live

Columns and tables support an optional expiration period called TTL (time-to-live); TTL is not supported on counter columns. Define the TTL value in seconds. Data expires once it exceeds the TTL period and is then marked with a tombstone. Expired data continues to be available for read requests during the grace period, see `gc_grace_seconds`. Normal compaction and repair processes automatically remove the tombstone data.

Note:

- TTL precision is one second, which is calculated by the coordinator node. When using TTL, ensure that all nodes in the cluster have synchronized clocks.
- A very short TTL is not very useful.
- Expiring data uses additional 8 bytes of memory and disk space to record the TTL and grace period.

WARNING:

The database storage engine can only encode TTL timestamps through January 19 2038 03:14:07 UTC due to the Year 2038 problem. The TTL date overflow policy determines whether requests with expiration timestamps later than the maximum date are rejected or inserted.

Setting a TTL for a specific column

```
CREATE KEYSPACE IF NOT EXISTS cycling WITH replication = {'class':'SimpleStrategy', 'replication_factor' : 1};
USE cycling;
```

```
CREATE TABLE cyclist_name ( id UUID PRIMARY KEY, lastname text, firstnam
e text );
```

```
INSERT INTO cyclist_name (id, lastname, firstname) VALUES (41d01b63-244e
-435d-bd6d-5dc8a0addb8c, 'A', 'J') USING TTL 86400;
```

```
SELECT TTL (lastname), id, lastname, firstname from cyclist_name WHERE i
d = 41d01b63-244e-435d-bd6d-5dc8a0addb8c;
```

```

ttl(lastname) | id | lastname | first
name
-----+-----+-----+-----
-----
      86396 | 41d01b63-244e-435d-bd6d-5dc8a0addb8c |      A |
      J
(1 rows)

```

The time-to-live value can also be updated with the USING TTL keywords in an UPDATE command.

```

UPDATE cyclist_name USING TTL 300 SET lastname = 'dummy' WHERE id = 41d01b63-244e-435d-bd6d-5dc8a0addb8c;

```

```

SELECT TTL (lastname), id, lastname, firstname from cyclist_name WHERE id = 41d01b63-244e-435d-bd6d-5dc8a0addb8c;

```

```

ttl(lastname) | id | lastname | first
name
-----+-----+-----+-----
-----
      274 | 41d01b63-244e-435d-bd6d-5dc8a0addb8c |    dummy |
      J
(1 rows)

```

To change the TTL of a specific column, you must re-insert the data with a new TTL. The database upserts the column with the new TTL. To remove TTL from a column, set TTL to zero.

Setting a TTL for a table

Use CREATE TABLE or ALTER TABLE to define the default_time_to_live property for all columns in a table. If any column exceeds TTL, the entire table is tombstoned.

```

ALTER TABLE cyclist_name WITH default_time_to_live = 10;

```

Insert data without TTL

```
INSERT INTO cyclist_name (id, lastname, firstname) VALUES (41d01b63-244e-435d-bd6d-5dc8a0addb8c, 'A', 'J');
```

```
SELECT TTL (lastname), id, lastname, firstname from cyclist_name WHERE id = 41d01b63-244e-435d-bd6d-5dc8a0addb8c;
```

ttl(lastname)	id	lastname	first name
---------------	----	----------	------------

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

9	41d01b63-244e-435d-bd6d-5dc8a0addb8c	A	J
---	--------------------------------------	---	---

(1 rows)