Multi-row Partitions

Paul O'Fallon

@paulofallon

Summary

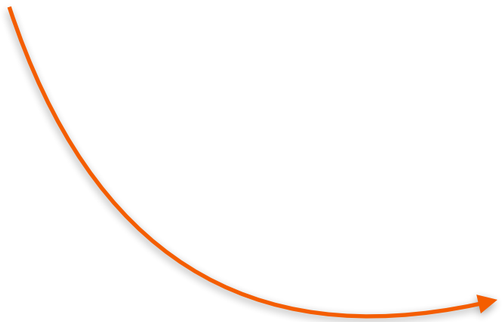
Composite keys

Static columns

Time series data

So Far...

```
CREATE TABLE courses (
  id varchar PRIMARY KEY
);
```



```
CREATE TABLE courses (
  id varchar,
  PRIMARY KEY (id)
);
```

Keys, Keys and More Keys!

```
PRIMARY KEY (id)
```


```
PRIMARY KEY (partition_key)
```

```
PRIMARY KEY (partition_key, clustering_key)
```

composite key

Composite Primary Keys

PRIMARY KEY (*p_key*, *c_key*)

PRIMARY KEY (*p_key*, c_key_1, … c_key_N)

Composite Primary Keys

```
PRIMARY KEY (p_key, c_key)

PRIMARY KEY (p_key, c_key1, … c_keyN)

PRIMARY KEY ((p_key1, … p_keyN), c_key1, … c_keyN)
```

Composite Primary Keys

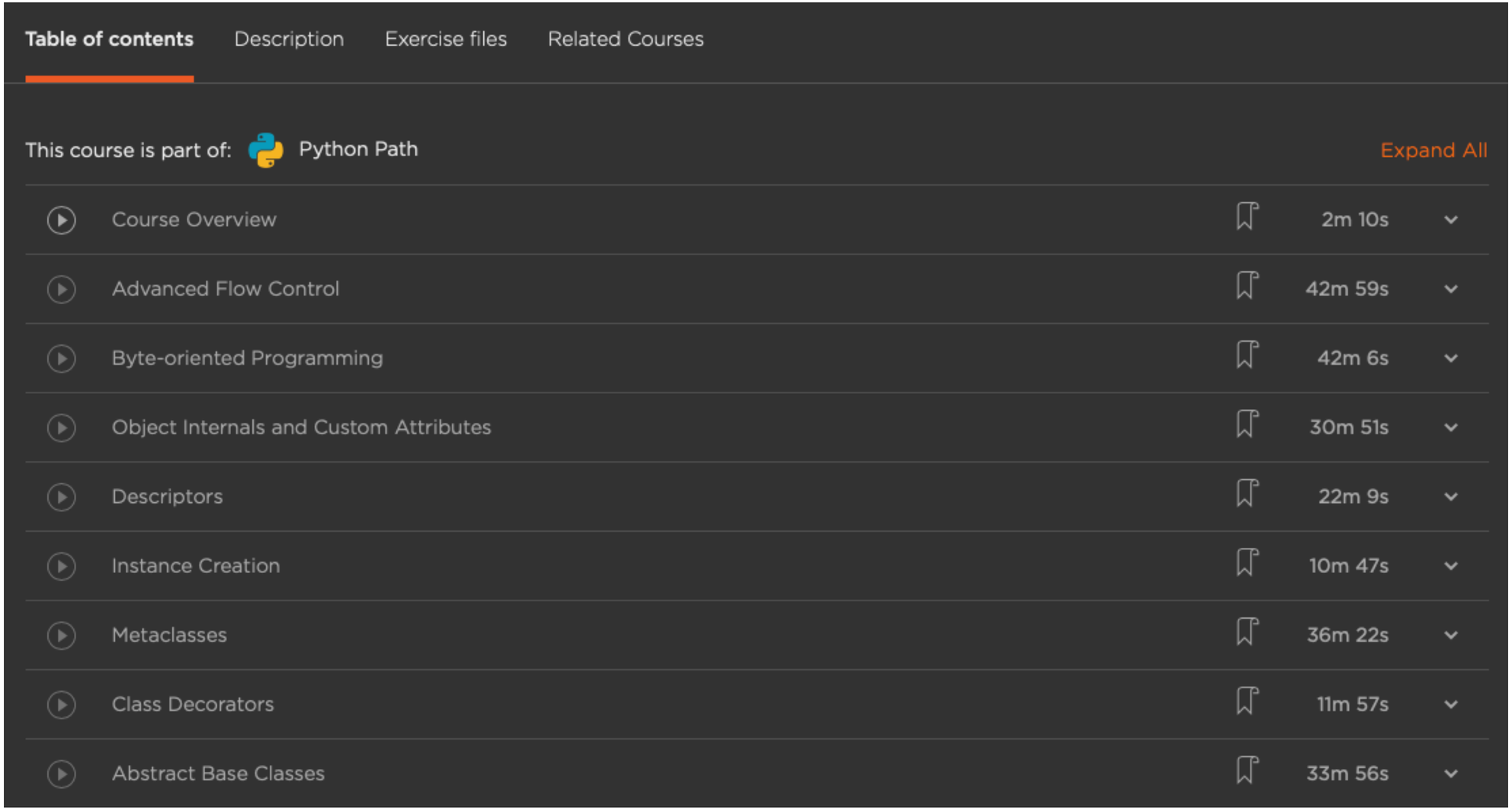
```
PRIMARY KEY (p_key, c_key)

PRIMARY KEY (p_key, c_key1, … c_keyN)

PRIMARY KEY ((p_key1, … p_keyN), c_key1, … c_keyN)

PRIMARY KEY ((p_key1, … p_keyN))
```

Courses, Now with Modules!



Revisiting Our Courses Schema

```
CREATE TABLE courses (
  id varchar,
  name varchar,
  PRIMARY KEY (id)
);
```

```
CREATE TABLE courses (
  id varchar,
  name varchar,
  module_id int,
  module_name varchar,
  PRIMARY KEY (id, module_id)
);
```

Revisiting Our Courses Schema

Inserting

```
INSERT INTO courses (id, name, module_id, module_name)
VALUES (
  ‘advanced-python’,'Advanced Python', 1,
  ‘Course Overview');

INSERT INTO courses (id, name, module_id, module_name)
VALUES (
  ‘advanced-python’,'Advanced Python', 2,
  ‘Advanced Flow Control’);
```

Revisiting Our Courses Schema

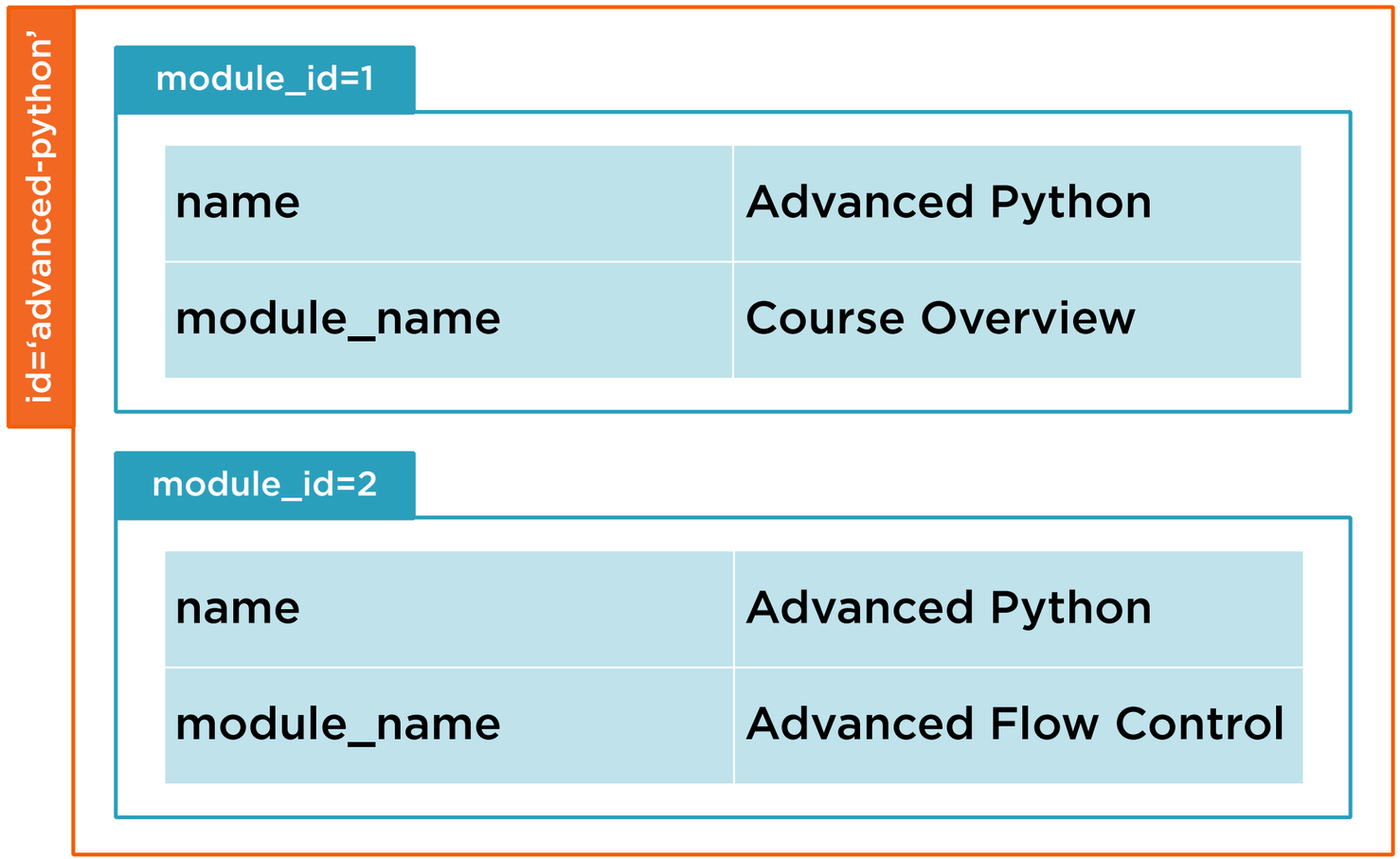
Selecting

```
SELECT * FROM courses WHERE id = ‘advanced-python';
```
← two rows

```
SELECT * FROM courses
WHERE id = ‘advanced-python' AND module_id = 1;
```
← one row

```
SELECT * FROM courses WHERE id = ‘advanced-python’
ORDER BY module_id DESC;
```
← two rows

Ordering By Clustering Key



Demo

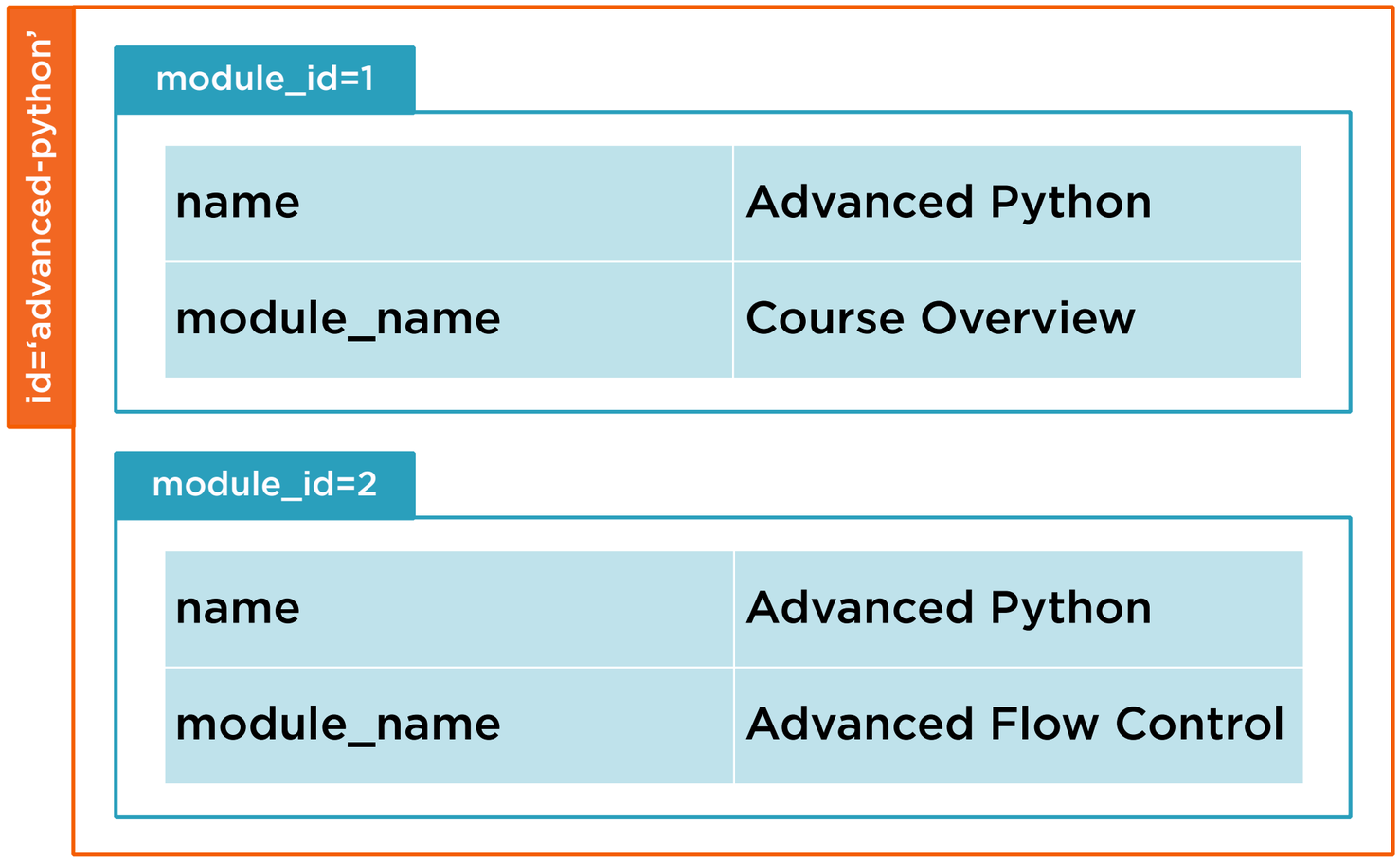
Add modules to our courses schema

Leverage a clustering key

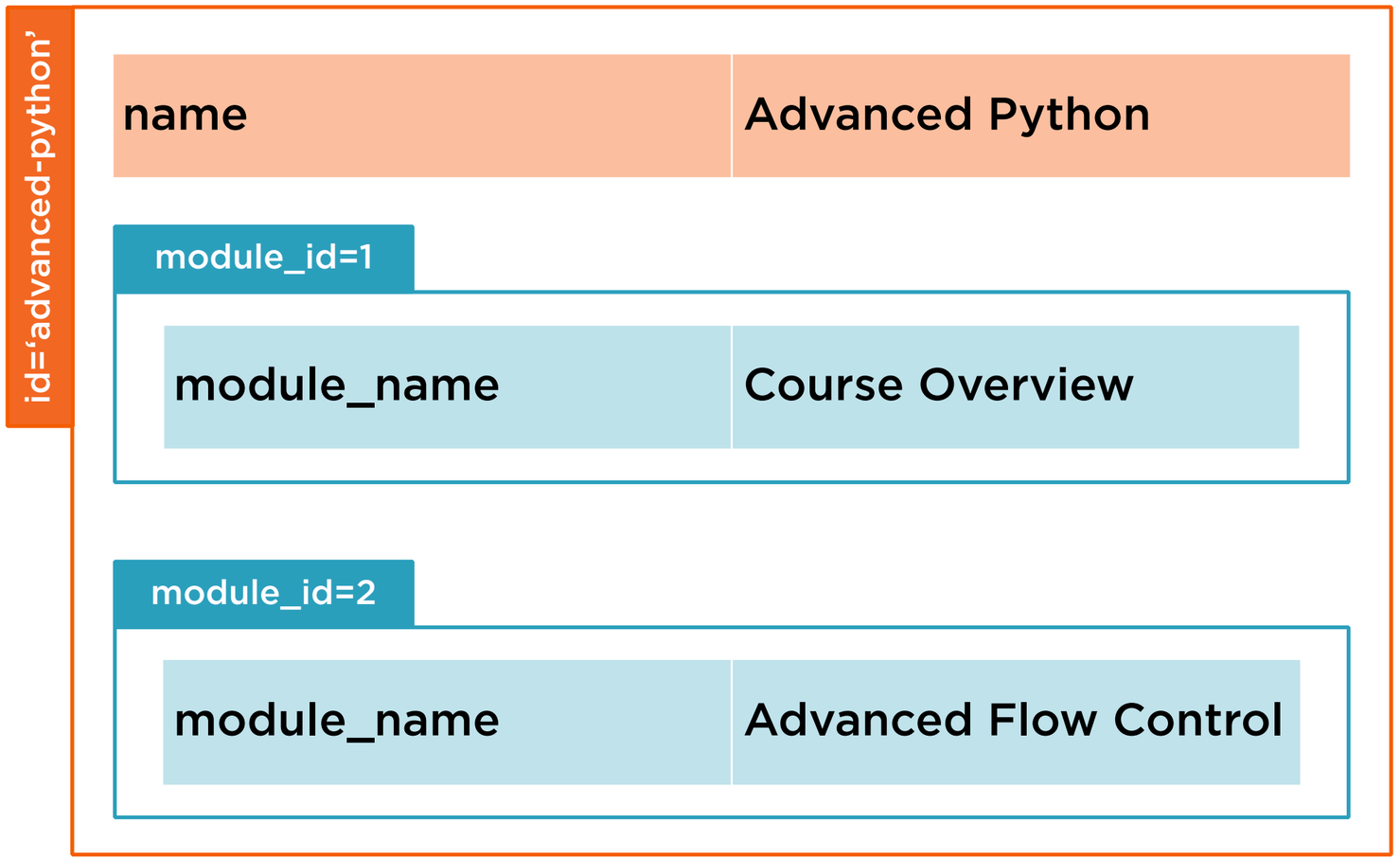
Static Columns

```
CREATE TABLE courses (
  id varchar,
  name varchar STATIC,
  module_id int,
  module_name varchar,
  PRIMARY KEY (id, module_id)
);
```

Without Static Columns



Static Columns



Adding Data with Static Columns

```
INSERT INTO courses (id, name)
VALUES (‘advanced-python’,’Advanced Python');

INSERT INTO courses (id, module_id, module_name)
VALUES (‘advanced-python', 1, ‘Course Overview');

UPDATE courses
SET module_name=‘Advanced Flow Control'
WHERE id=‘advanced-python’ AND module_id=2;
```

SELECT REMAINS THE SAME

Demo

Convert course fields to static columns

Select course and module-level data

Time Series Data

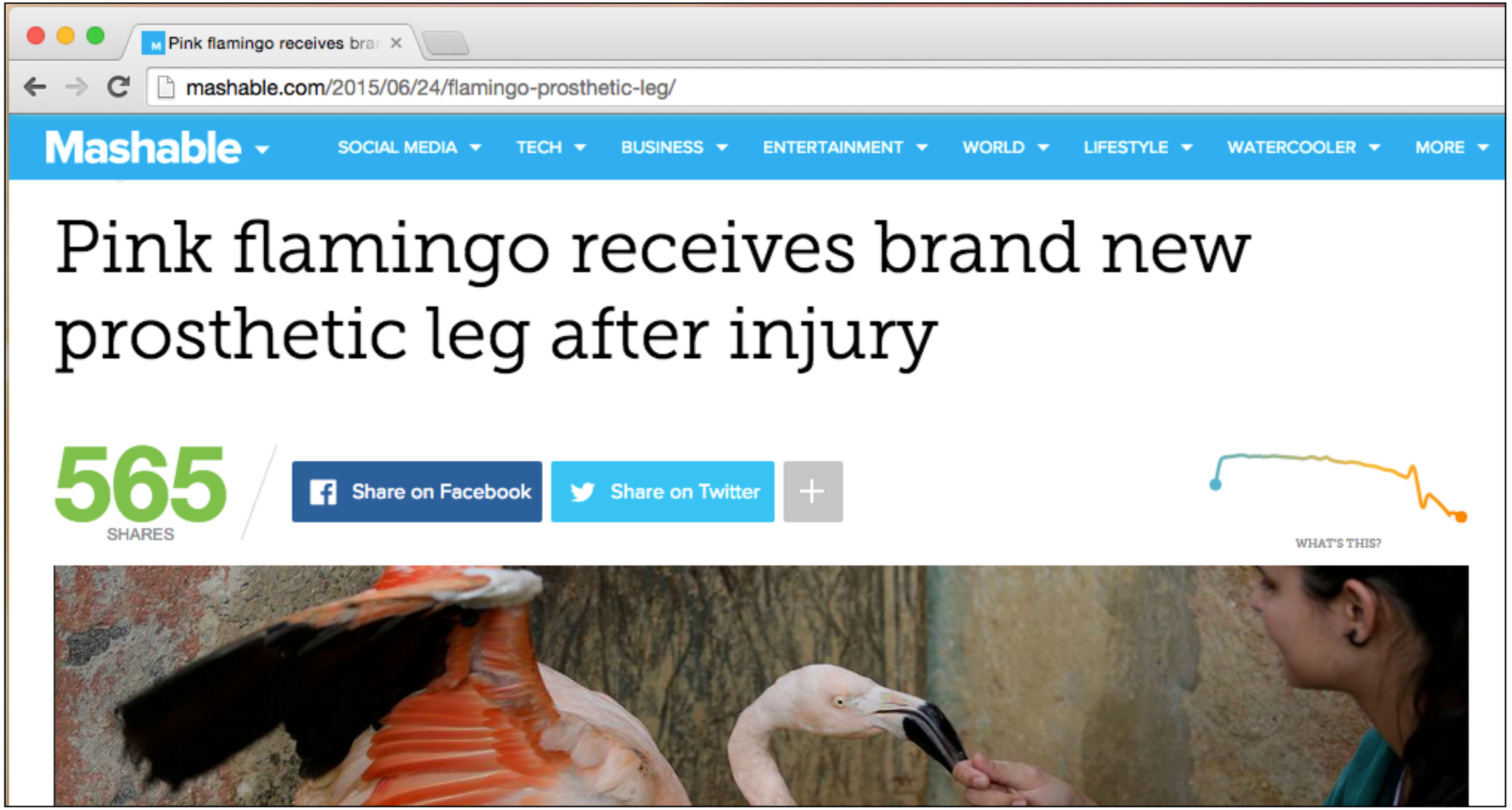
TimeUUID Data Type

`45b94a50-12e5-11e5-9114-091830ac5256`

Version 1 UUID comprised of:

- The number of 100 ns intervals since UUID epoch
- MAC address
- Clock sequence number to prevent duplicates

TimeUUID Use Case



TimeUUID Use Case

```
CREATE TABLE course_page_views (
    course_id text,
    view_id timeuuid,
    PRIMARY KEY (course_id, view_id)
) WITH CLUSTERING ORDER BY (view_id DESC);
```

TimeUUID Functions

now

```
INSERT INTO course_page_views (course_id, view_id)
VALUES (‘advanced-python‘, now());
```

dateOf / unixTimestampOf

```
SELECT course_id, dateOf(view_id)
FROM course_page_views WHERE course_id = ‘advanced-python‘;
```

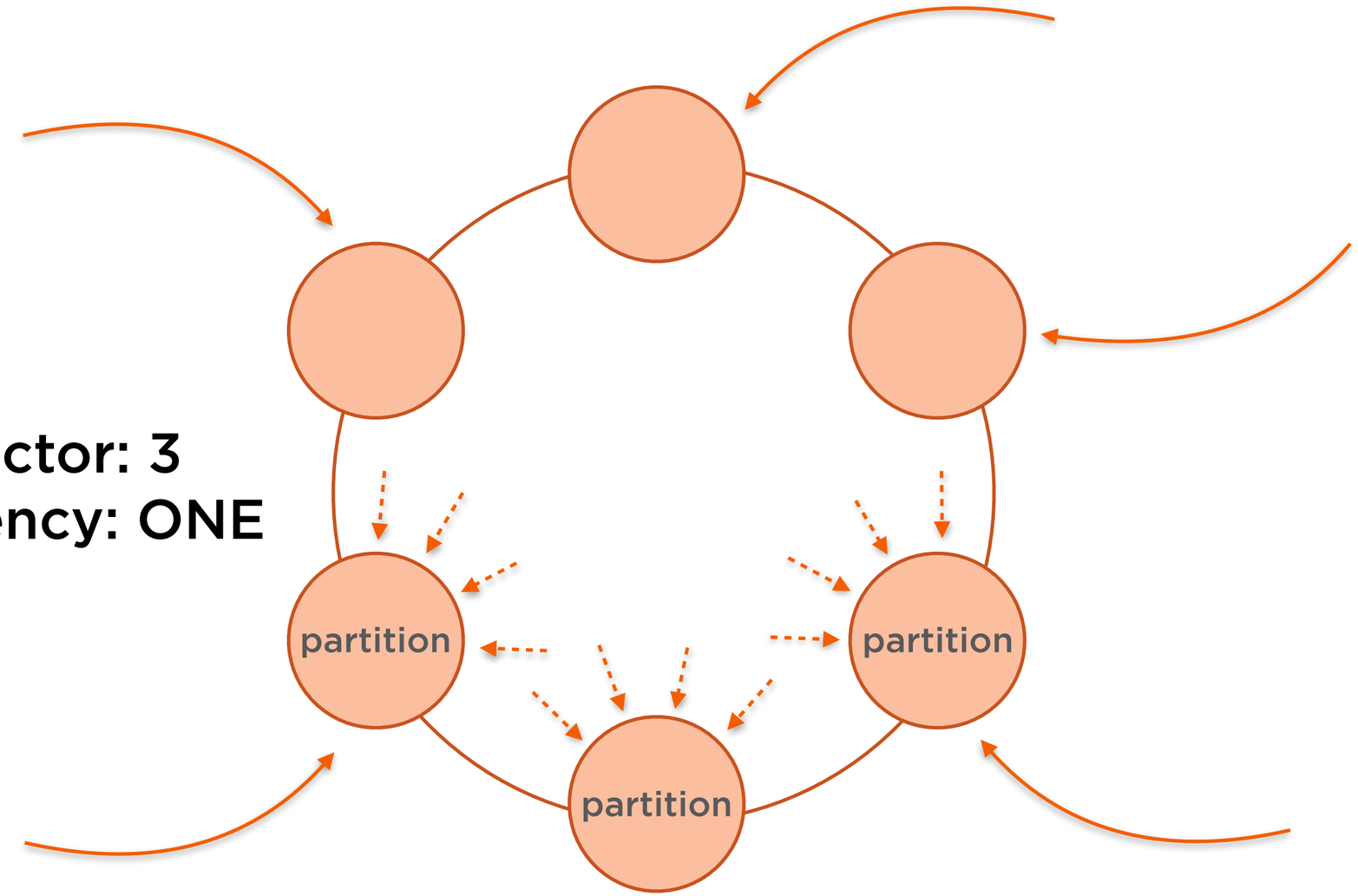
TimeUUID Functions

minTimeuuid / maxTimeuuid

```
SELECT dateOf(view_id)
FROM course_page_views
WHERE course_id = ‘advanced-python'
AND view_id >= maxTimeuuid('2019-11-01 00:00+0000')
AND view_id < minTimeuuid(‘2019-12-01 00:00+0000')
```

Storing Data with a Clustering Key

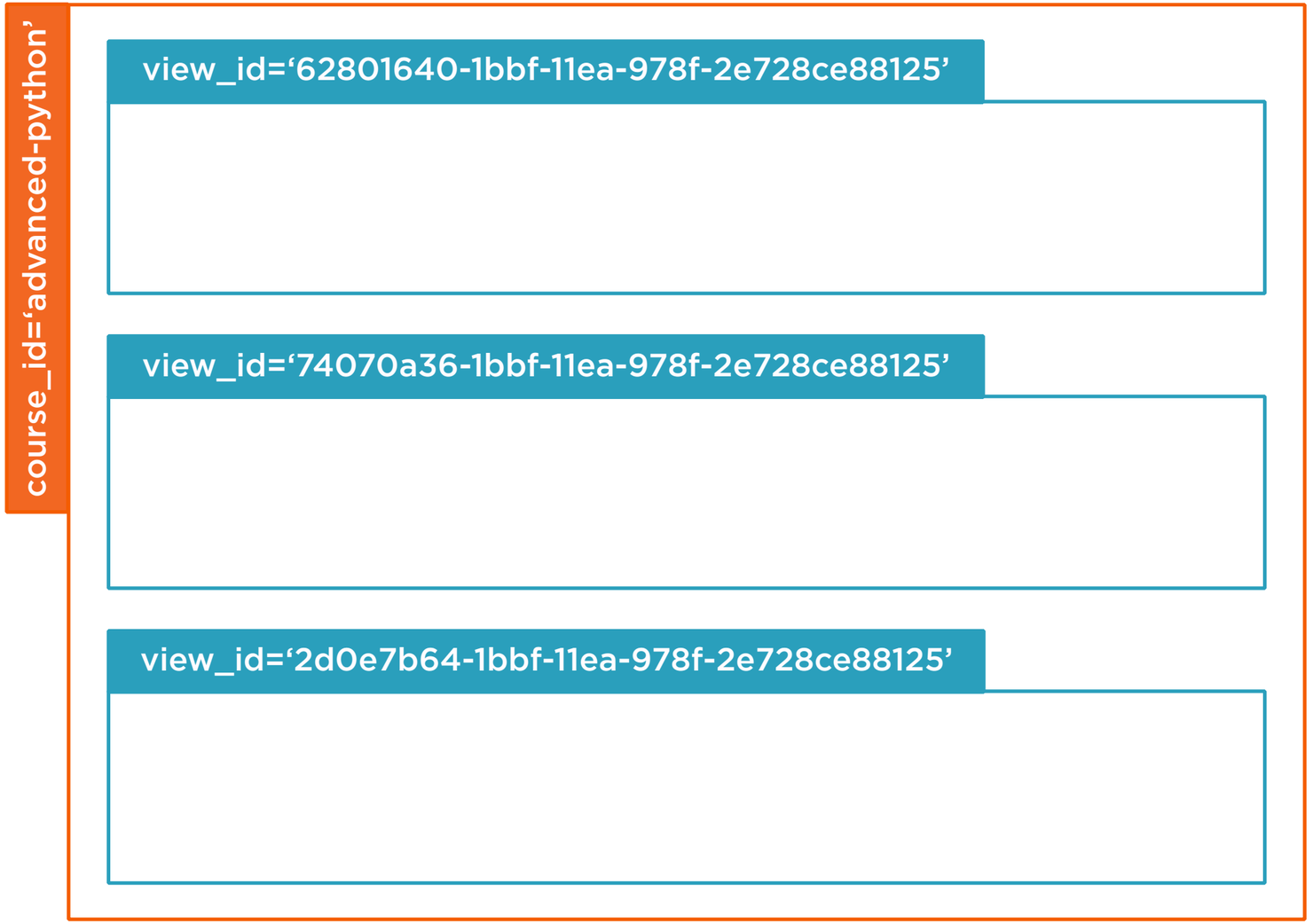
replication_factor: 3
write consistency: ONE



Storing Data with a Clustering Key

Storing Data with a Clustering Key

Storing Data with a Clustering Key



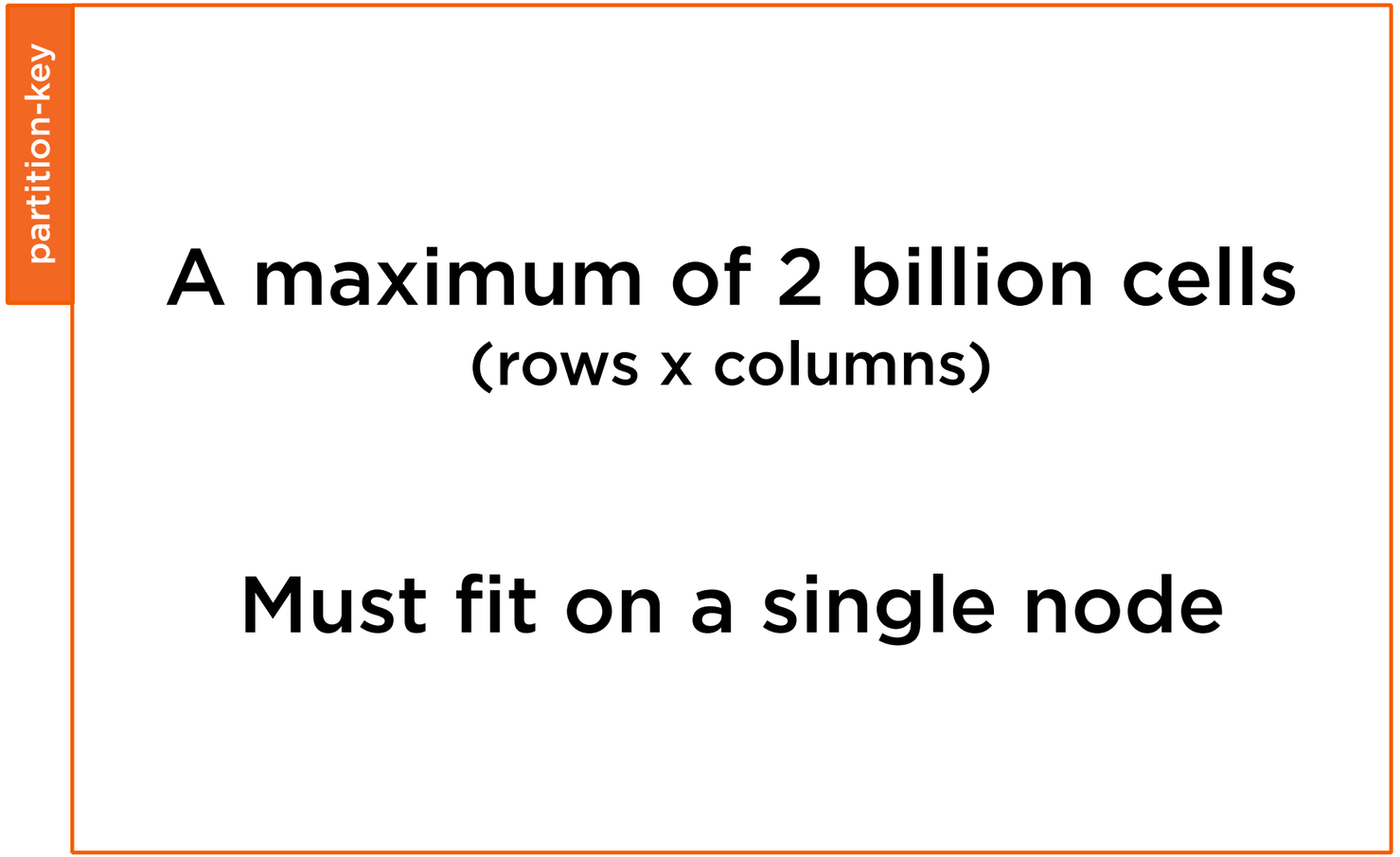
Demo

Create a table for course page views

Insert data with a TimeUUID and TTL

Track the last time a page was viewed

Bucketing Time Series Data



Bucketing Time Series Data

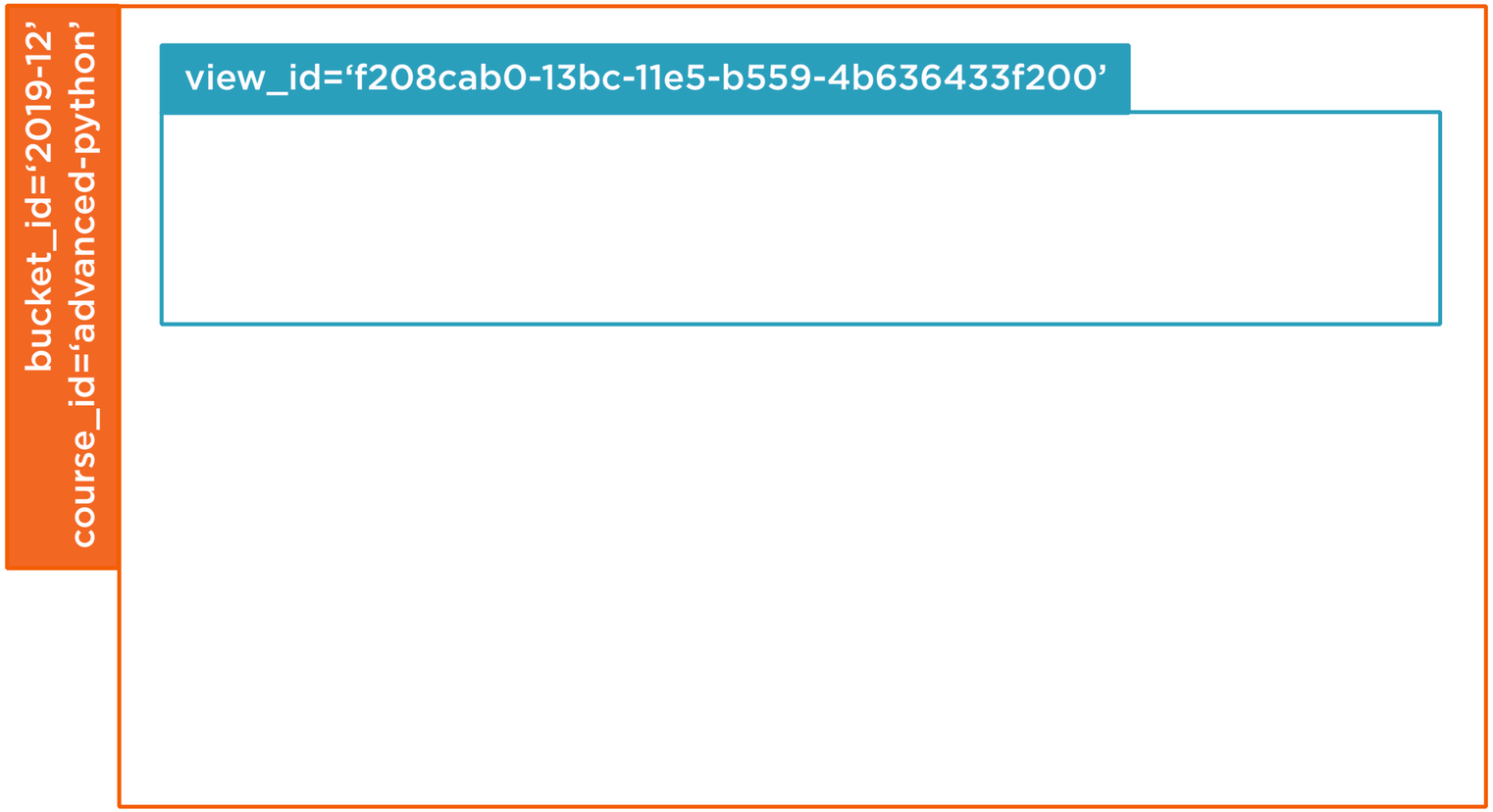
Bucketing Use Case

```
CREATE TABLE course_page_views (
    bucket_id text,
    course_id text,
    view_id timeuuid,
    PRIMARY KEY ((bucket_id, course_id), view_id)
) WITH CLUSTERING ORDER BY (view_id DESC);
```

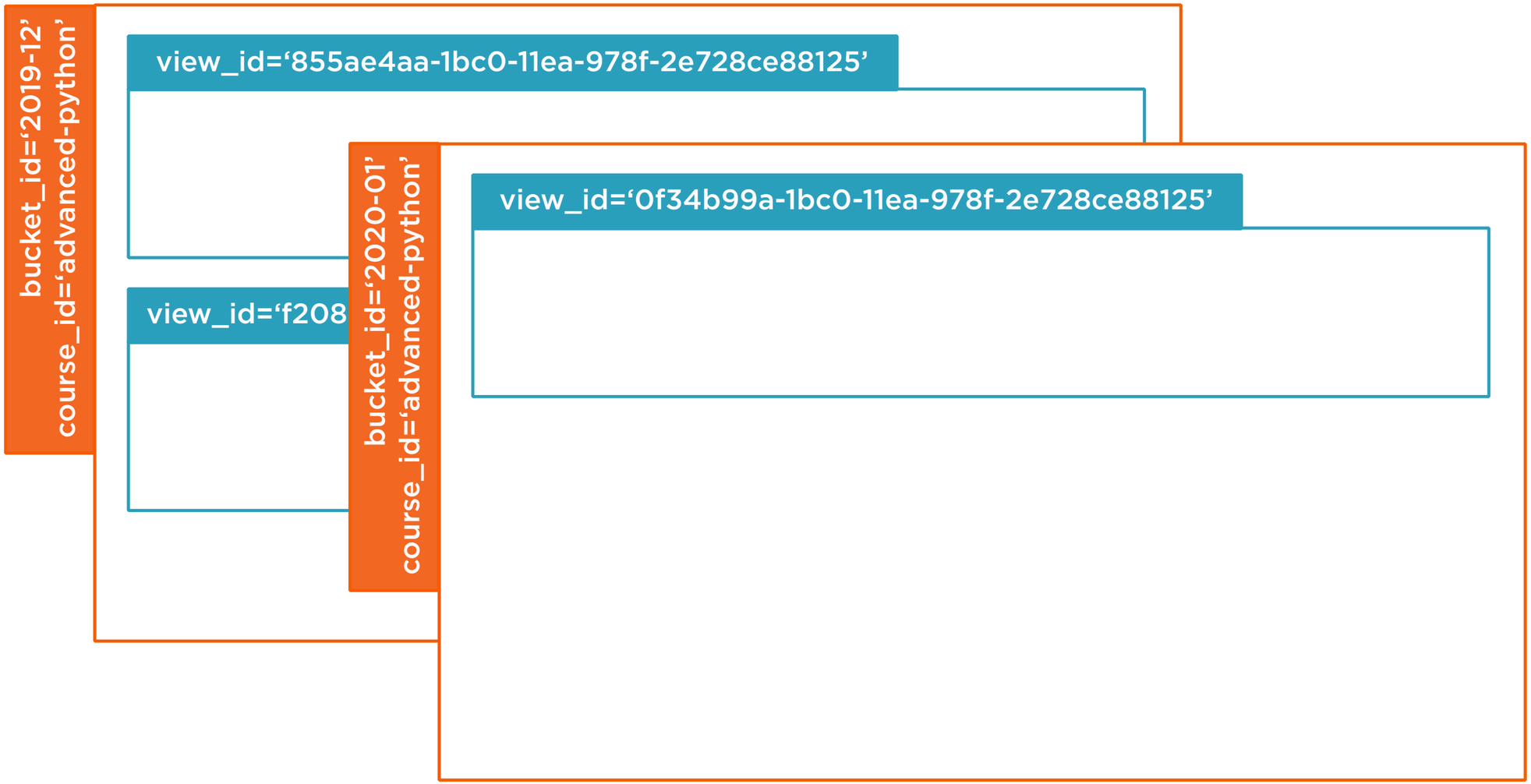
Inserting Bucketed Data

```
INSERT INTO course_page_views
  (bucket_id, course_id, view_id)
VALUES
  ('2019-12', 'advanced-python', now());
```

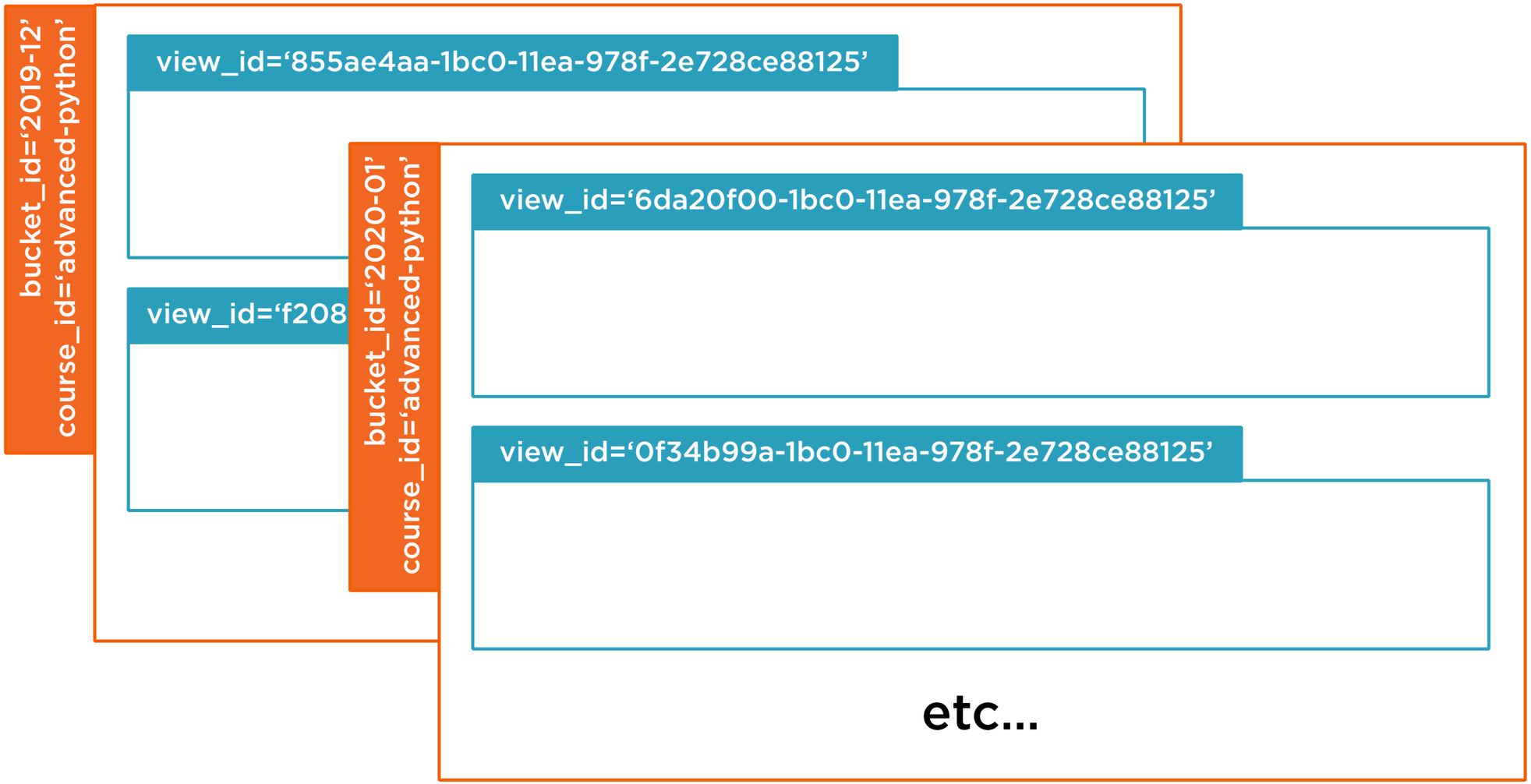
Inserting Bucketed Data



Inserting Bucketed Data



Inserting Bucketed Data



Demo

Use a bucket id to segment page views

Conclusion

Clustering and composite primary keys

Static columns

Time series data

TimeUUID data type

TTL vs. bucketing