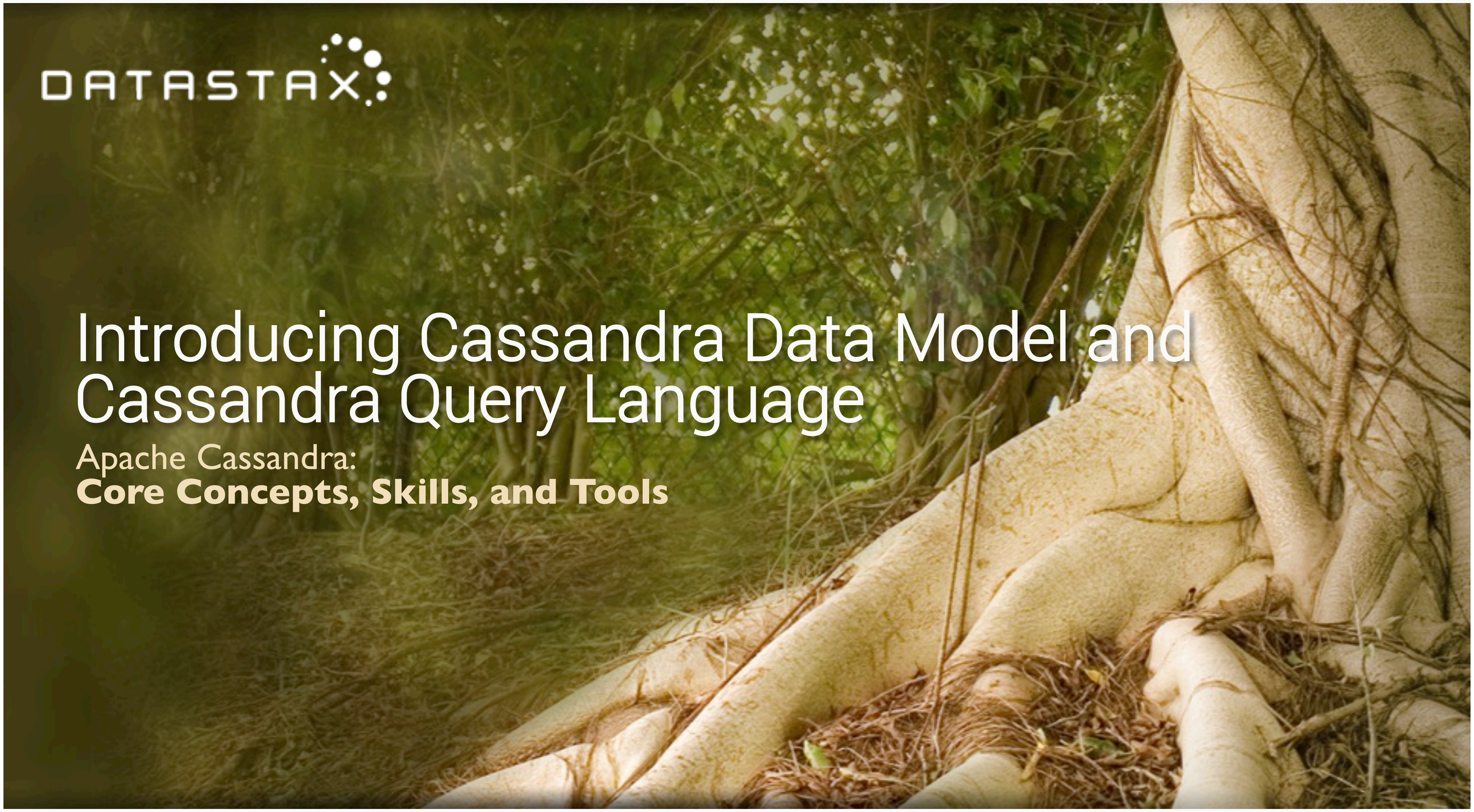


Introducing Cassandra Data Model and Cassandra Query Language

Apache Cassandra:
Core Concepts, Skills, and Tools

Artem Chebotko, Leo Schuman

Learning Objectives

- **Understand the Cassandra data model**
- Introduce *cqlsh* (optional)
- Understand and use the DDL subset of CQL
- Introduce *DevCenter*
- Understand and use the DML subset of CQL
- Understand basics of data modeling (optional)

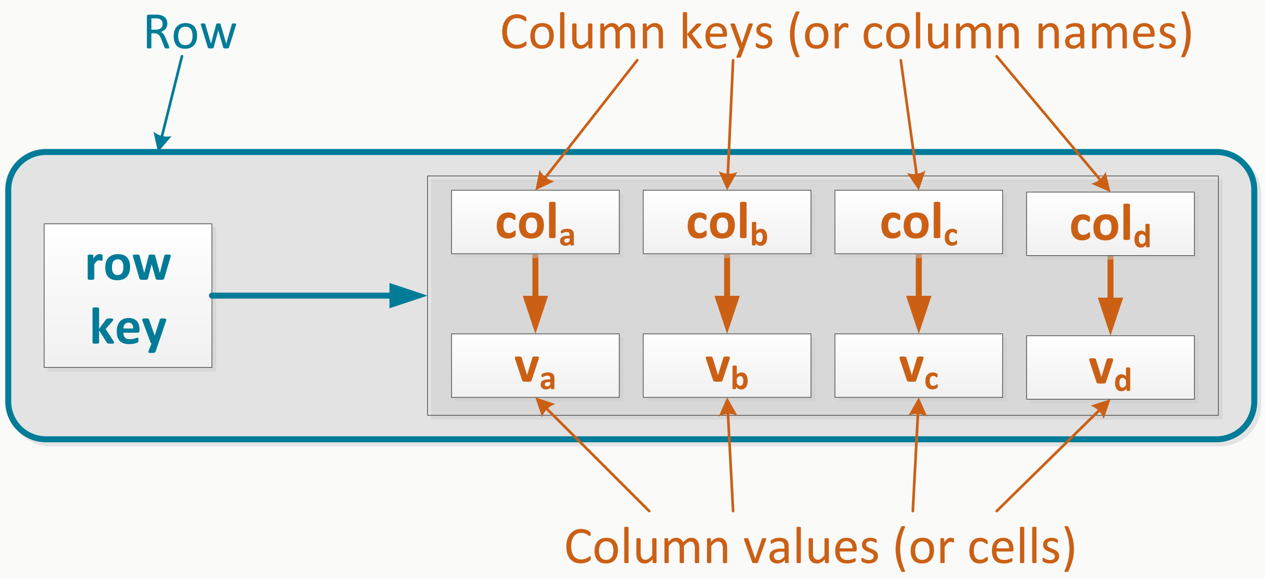
What are the essential constituents of the Cassandra data model?

- The Cassandra data model defines
  1. *Column family* as a way to store and organize data
  2. *Table* as a two-dimensional view of a multi-dimensional *column family*
  3. Operations on tables using the Cassandra Query Language (CQL)

- We cover these three constituents in the order they are listed
  - Understanding *column families* is a prerequisite to understanding *tables*
  - Understanding *tables* is a prerequisite to understanding operations

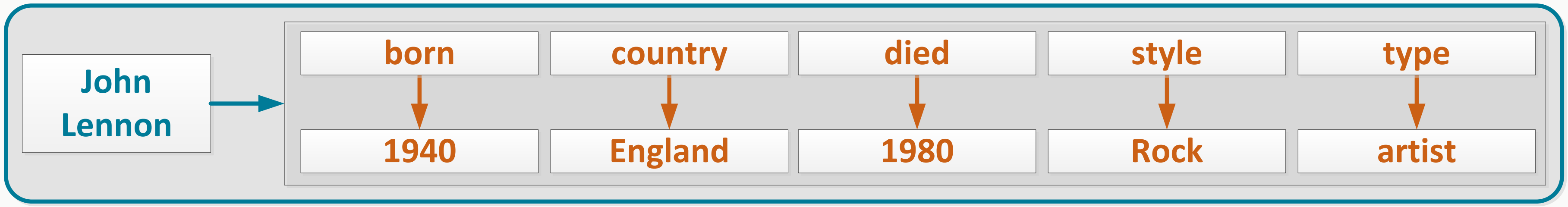
What are row, row key, column key, and column value?

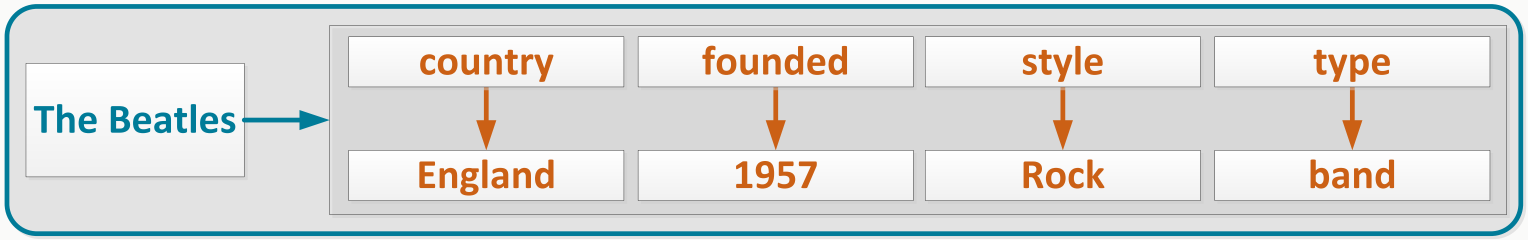
- *Row* is the smallest unit that stores related data in Cassandra
  - Rows – individual rows constitute a *column family*
  - Row key – uniquely identifies a *row* in a *column family*
  - Row – stores pairs of *column keys* and *column values*
  - Column key – uniquely identifies a *column value* in a *row*
  - Column value – stores one value or a *collection* of values



What are row, row key, column key, and column value?

- Sample rows that describe an artist and a band
  - *Column keys* are inherently sorted

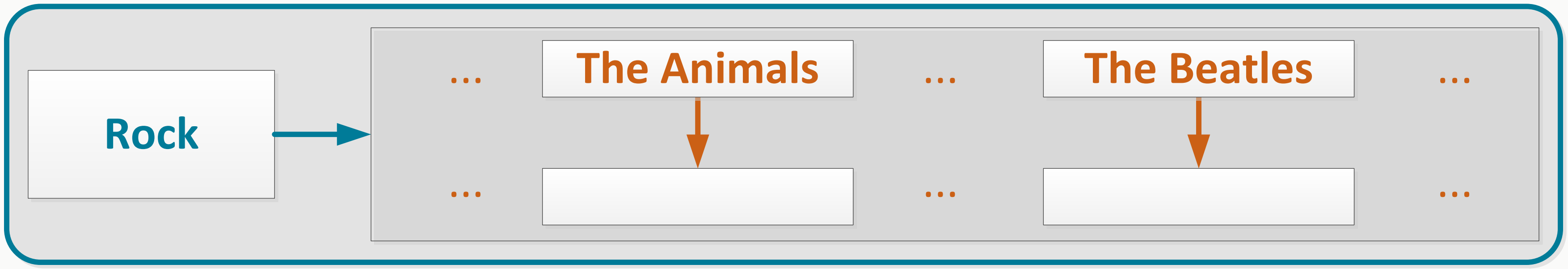




- A *row* can be retrieved if its *row key* is known
- A *column value* can be retrieved if its *row key* and *column key* are known

What is a wide row?

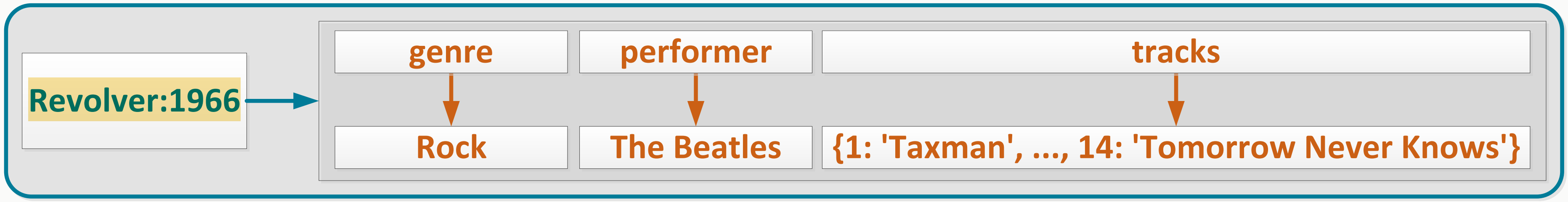
- Rows may be described as “skinny” or “wide”
  - Skinny row – has a fixed, relatively small number of *column keys*
    - Previous examples were skinny rows
  - Wide row – has a relatively large number of *column keys* (hundreds or thousands); this number may increase as new data values are inserted
    - For example, a row that stores all bands of the same style
    - The number of such bands will increase as new bands are formed



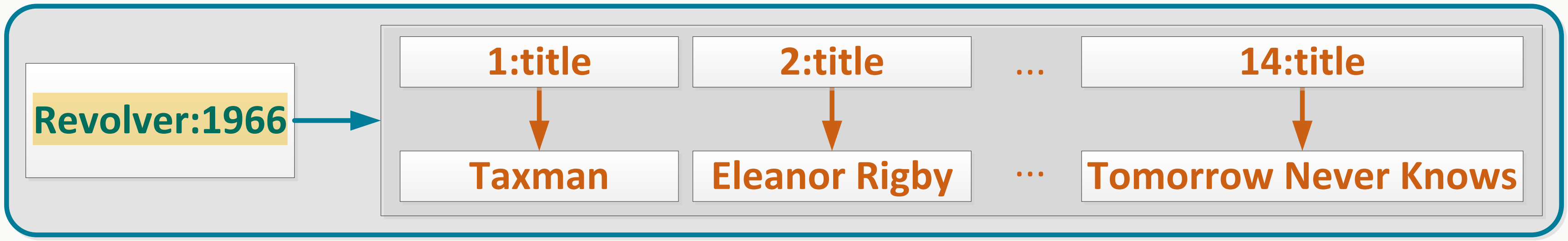
- Note that column values do not exist in this example
  - The column key – in this case a band name – stores all the data desired
  - Could have stored the number of albums, or year founded, etc., as column values

What are composite row key and composite column key?

- Composite row key – multiple components separated by colon



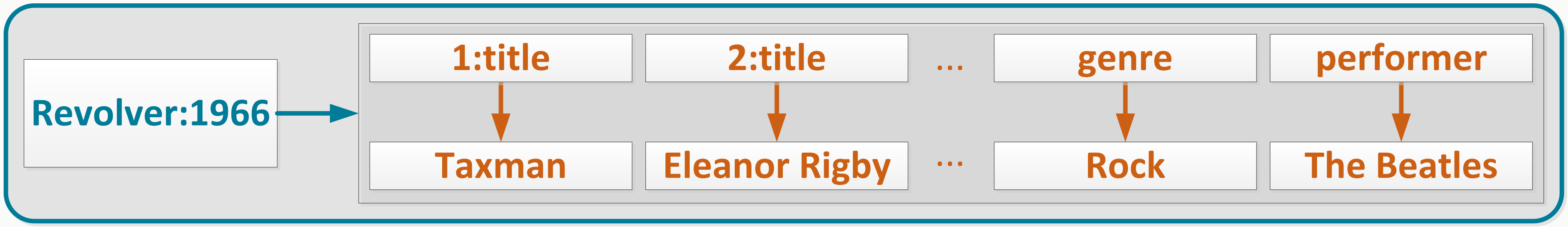
  - ‘Revolver’ and 1966 are the album title and year
  - ‘tracks’ value is a collection (map)
- Composite column key – multiple components separated by colon
  - Composite column keys are sorted by each component



  - 1,2, …, 14 are track numbers; ‘title’ is metadata
    - We could have stored actual title as components of composite column keys: 1:Taxman, 2:Eleanor Rigby, …, 14:Tomorrow Never Knows

Can simple and composite column keys co-exist in the same row?

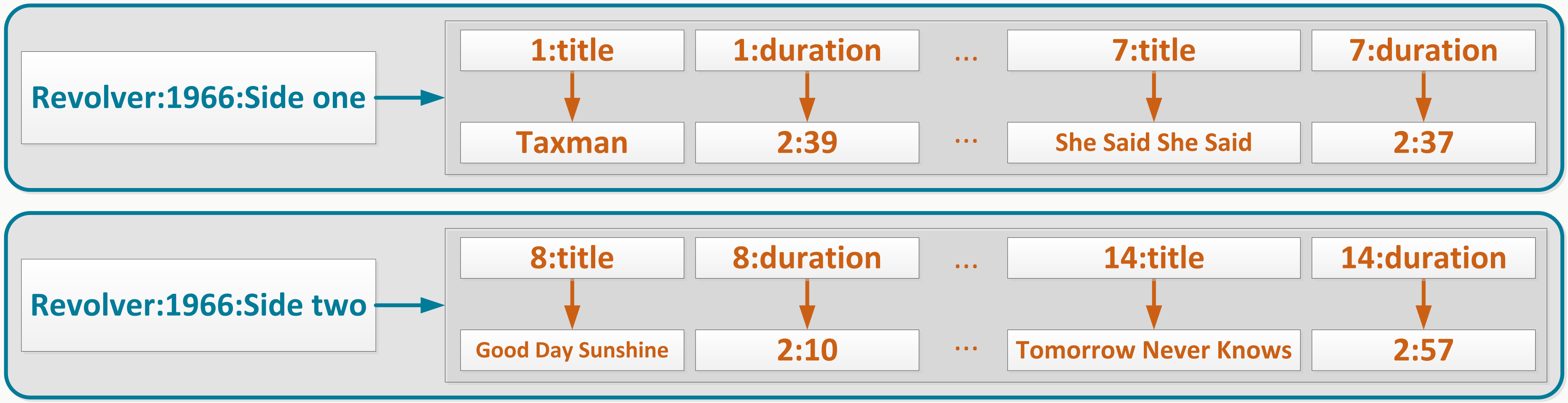
- Row can contain both simple and composite column keys



  - ‘genre’ and ‘performer’ are simple column keys
  - ‘1:title’, ‘2:title’, … are composite column keys

What components of a row can store useful values?

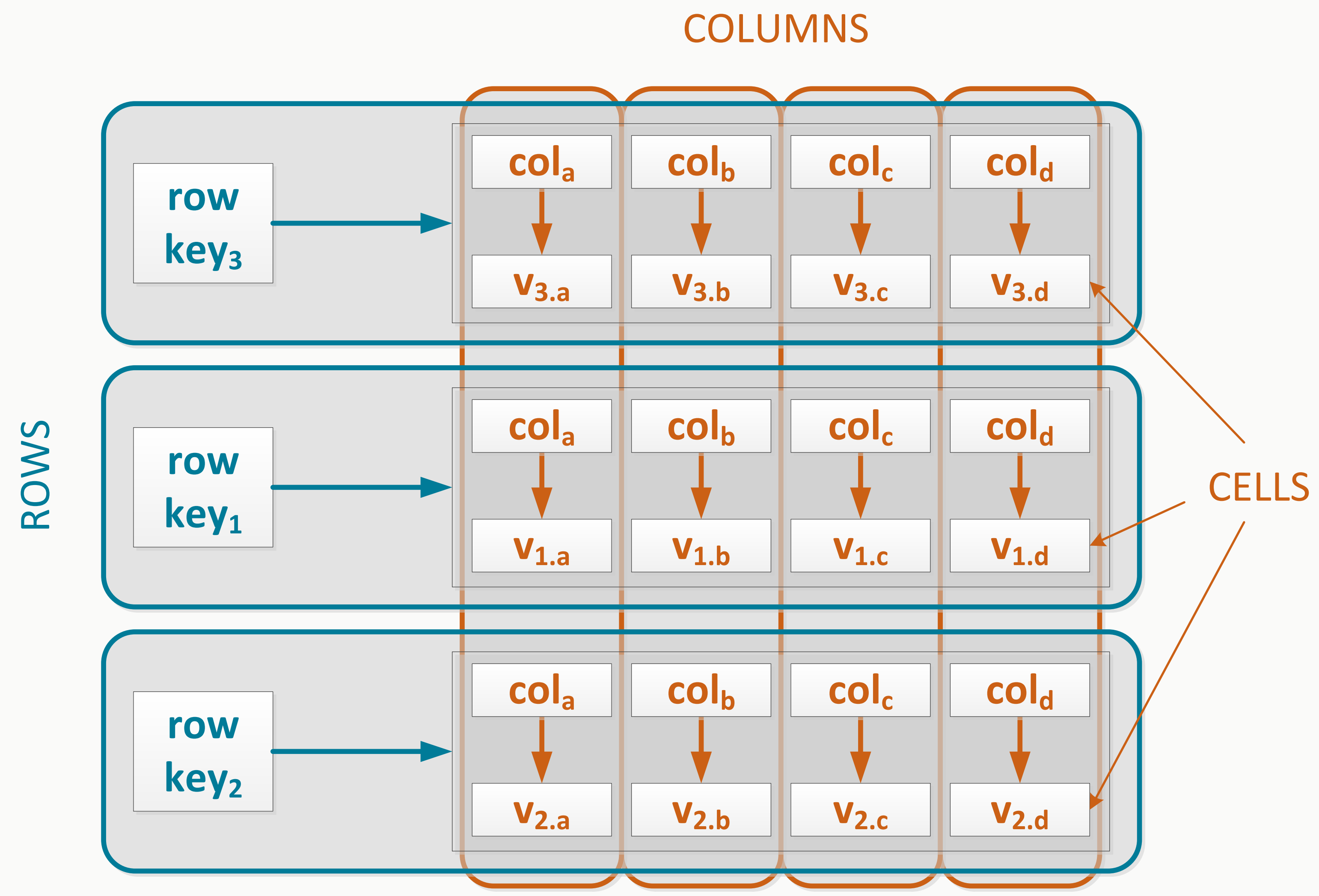
- Any component of a row can store *data* or *metadata*
  - Simple or composite row keys
  - Simple or composite column keys
  - Atomic or set-valued (collection) column values



- Metadata: 'Side one', 'Side two', 'title', 'duration'
- Data: everything else ('Revolver', '1966', 'She Said She Said', etc.)

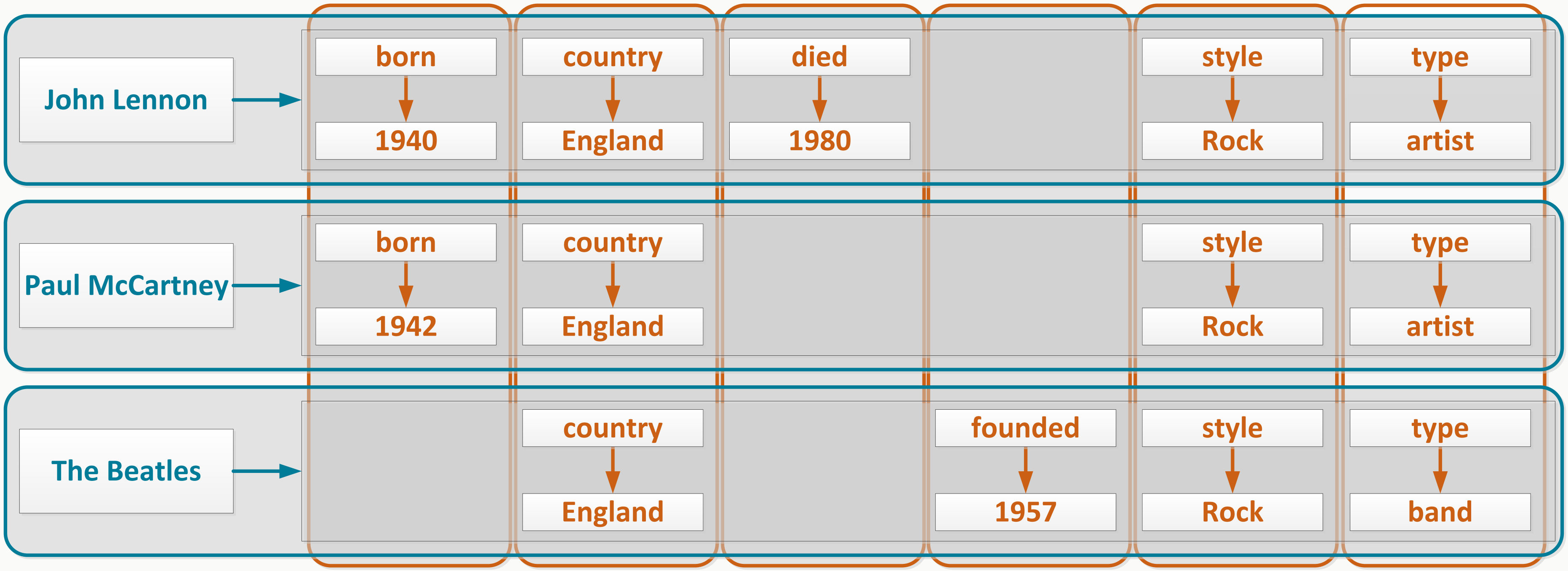
What is a column family?

- Column family – set of rows with a similar structure



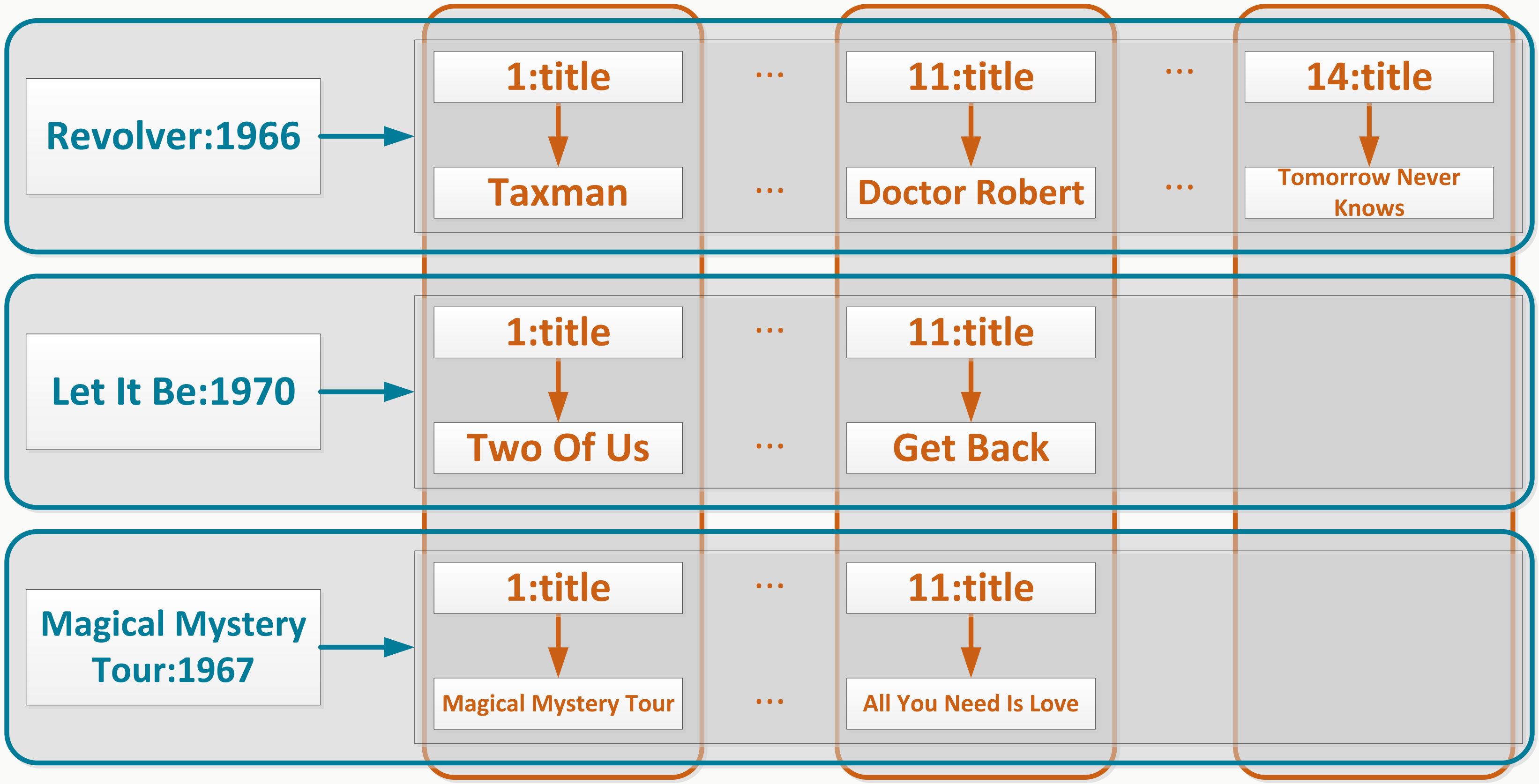
What is a column family?

- Distributed
- Sparse
    - Column family that stores data about artists and bands



What is a column family?

- Sorted columns
- Multidimensional
  - Column family that stores albums and their tracks



What are the size limitations for a column family?

- Size of a *column family* is only limited to the size of a *cluster*
  - Linear scalability
  - *Rows* are distributed among the *nodes* in a *cluster*
- *Column family* component size considerations
  - Data from a one row must fit on one node
    - Data from any given row never spans multiple nodes
  - Maximum columns per row is 2 billion
    - In practice – Up to 100 thousand
  - Maximum data size per cell (column value) is 2 GB
    - In practice – Up to 100 MB

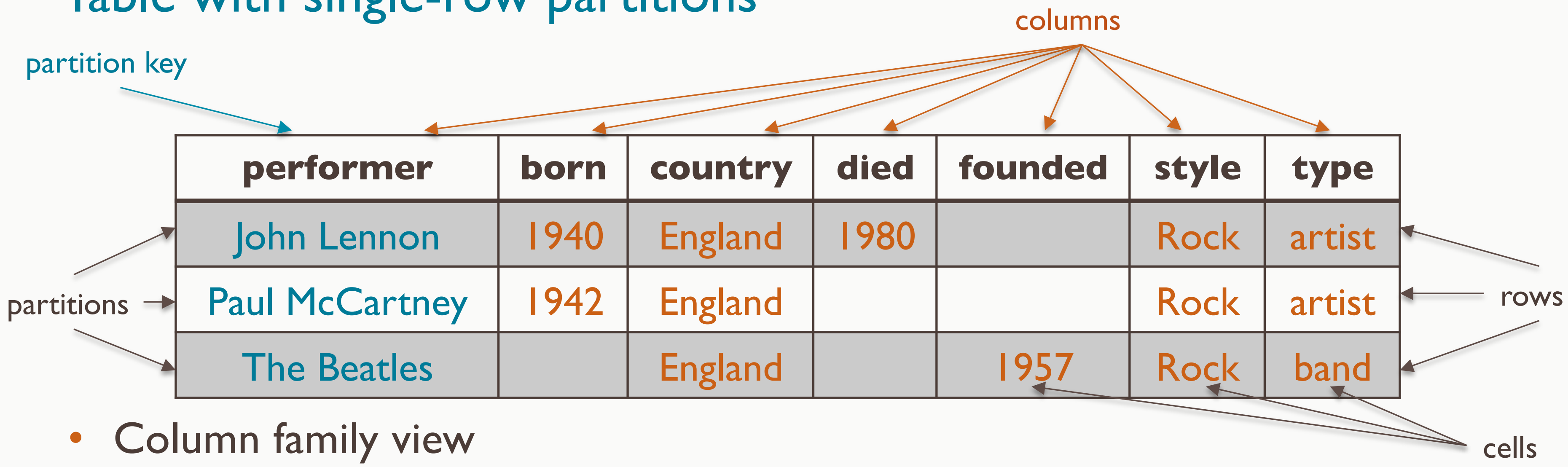
Exercise 1: Model sample data as column families

What is a CQL table and how is it related to a column family?

- A *CQL table* is a *column family*
  - CQL tables provide two-dimensional views of a column family, which contains potentially multi-dimensional data, due to composite keys and collections
- *CQL table* and *column family* are largely interchangeable terms
  - Not surprising when you recall *tables* and *relations*, *columns* and *attributes*, *rows* and *tuples* in relational databases
- Supported by declarative language Cassandra Query Language
  - Data Definition Language, subset of CQL
  - SQL-like syntax, but with somewhat different semantics
  - Convenient for defining and expressing Cassandra database schemas

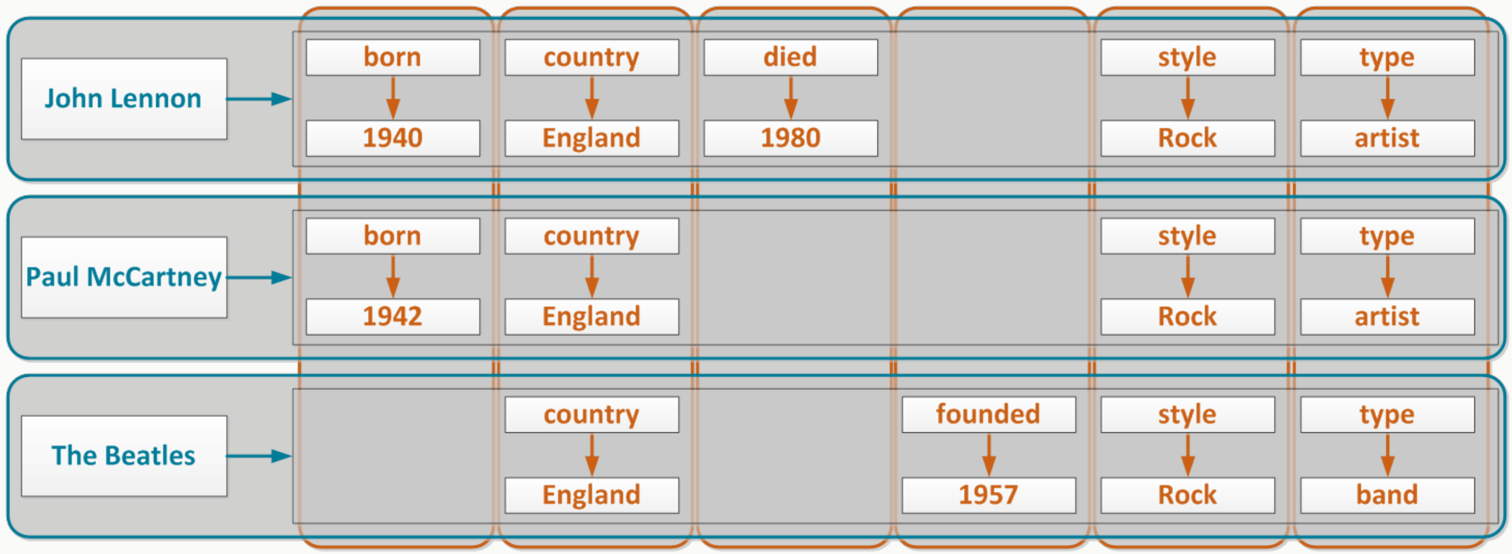
What are partition, partition key, row, column, and cell?

- Table with single-row partitions



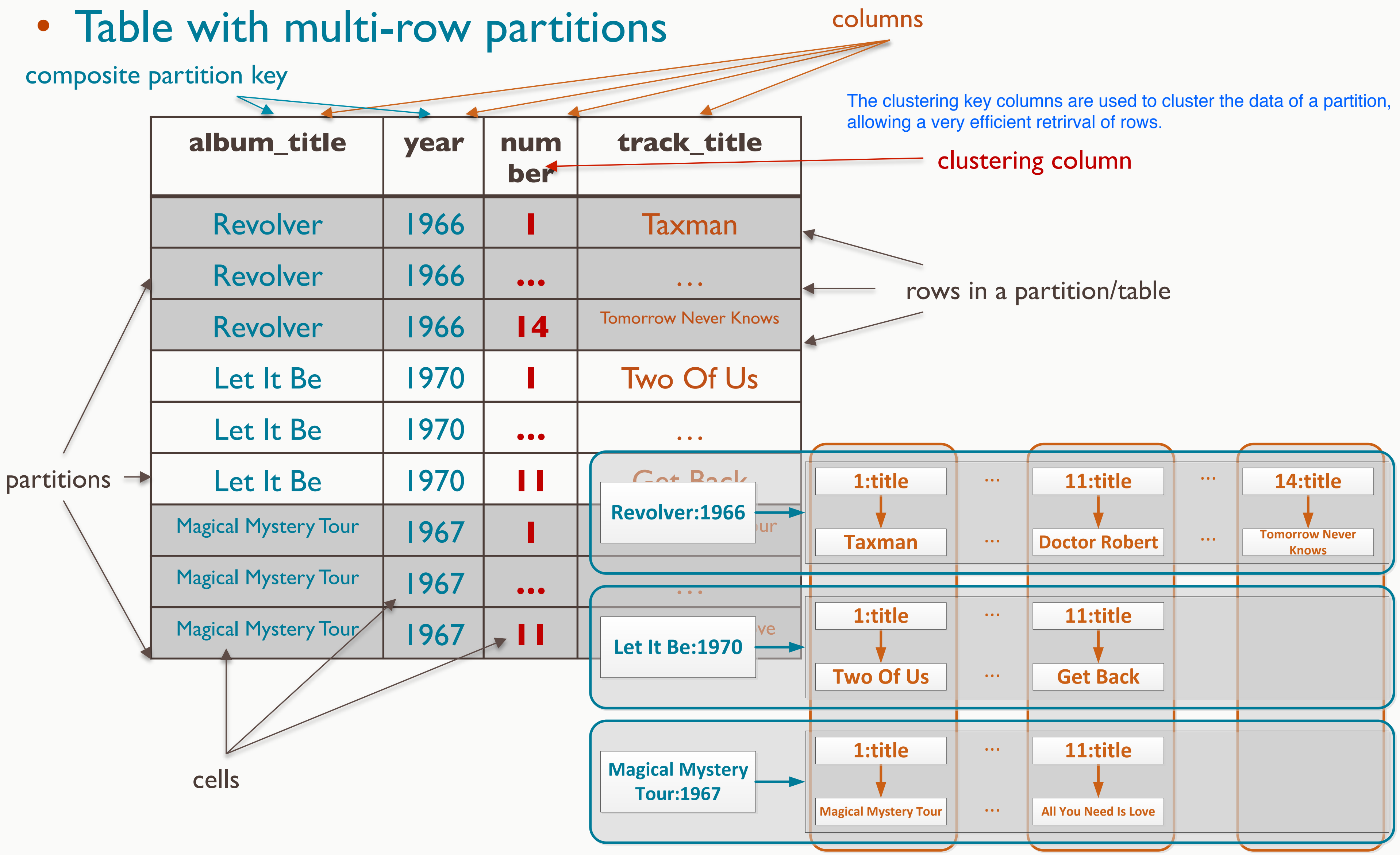
performer	born	country	died	founded	style	type
John Lennon	1940	England	1980		Rock	artist
Paul McCartney	1942	England			Rock	artist
The Beatles		England		1957	Rock	band

- Column family view



What are composite partition key and clustering column?

- Table with multi-row partitions



album_title	year	number	track_title
Revolver	1966	1	Taxman
Revolver	1966	...	...
Revolver	1966	14	Tomorrow Never Knows
Let It Be	1970	1	Two Of Us
Let It Be	1970	...	...
Let It Be	1970	11	Get Back
Magical Mystery Tour	1967	1	ur
Magical Mystery Tour	1967	...	...
Magical Mystery Tour	1967	11	ve

What are static columns?

- Table with multi-row partitions

composite partition key

clustering column

static columns

album_title	year	number	genre	performer	track_title
Revolver	1966	1	Rock	The Beatles	Taxman
Revolver	1966	…	Rock	The Beatles	…
Revolver	1966	14	Rock	The Beatles	Tomorrow Never Knows
Let It Be	1970	1	Rock	The Beatles	Two Of Us
Let It Be	1970	…	Rock	The Beatles	…
Let It Be	1970	11	Rock	The Beatles	Get Back
Magical Mystery Tour	1967	1	Rock	The Beatles	Magical Mystery Tour
Magical Mystery Tour	1967	…	Rock	The Beatles	…
Magical Mystery Tour	1967	11	Rock	The Beatles	All You Need Is Love

partitions

rows in a partition

cells

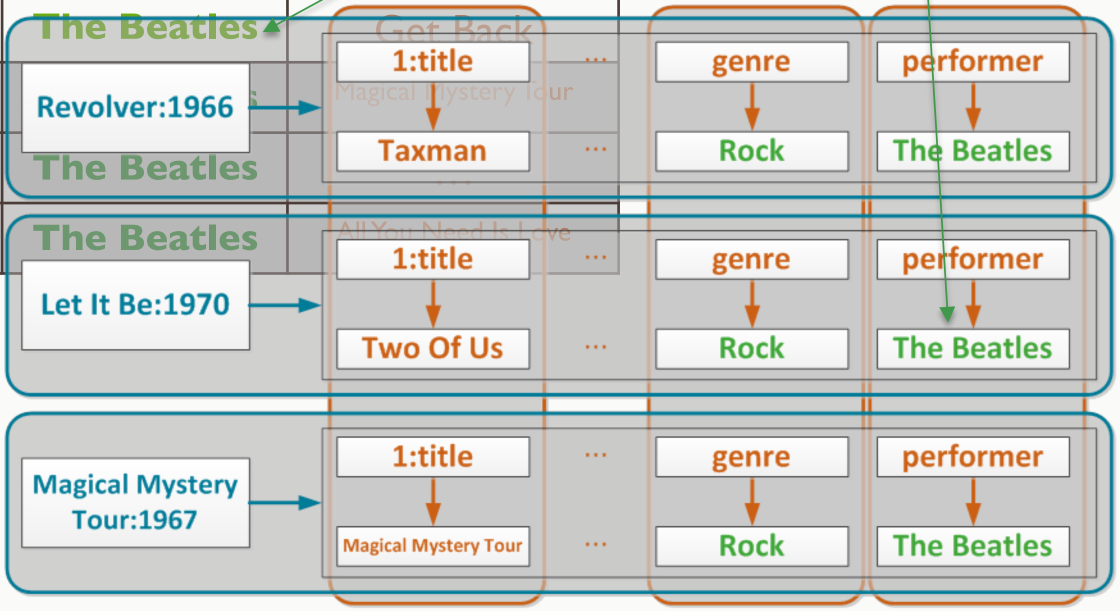
- Static column values are shared for all rows in a multi-row partition

What are static columns?

- Table with multi-row partitions

album_title	year	number	genre	performer	track_title
Revolver	1966	1	Rock	The Beatles	Taxman
Revolver	1966	…	Rock	The Beatles	…
Revolver	1966	14	Rock	The Beatles	Tomorrow Never Knows
Let It Be	1970	1	Rock	The Beatles	Two Of Us
Let It Be	1970	…	Rock	The Beatles	…
Let It Be	1970	11	Rock	The Beatles	Get Back
Magical Mystery Tour	1967	1	Rock	The Beatles	Magical Mystery Tour
Magical Mystery Tour	1967	…	Rock	The Beatles	…
Magical Mystery Tour	1967	11	Rock	The Beatles	All You Need Is Love

static column value



What is a primary key?

- Primary key uniquely identifies a row in a table
  - Simple or composite partition key and all clustering columns (if present)

performer	born	country	died	founded	style	type
John Lennon	1940	England	1980		Rock	artist
Paul McCartney	1942	England			Rock	artist
The Beatles		England		1957	Rock	band

- Primary key (table above)
  - performer
- Primary key (table below)
  - album, year, number

- Static columns cannot be part of a primary key

album_title	year	num ber	track_title
Revolver	1966	1	Taxman
Revolver	1966	...	...
Revolver	1966	14	Tomorrow Never Knows
Let It Be	1970	1	Two Of Us
Let It Be	1970	...	...
Let It Be	1970	11	Get Back
Magical Mystery Tour	1967	1	Magical Mystery Tour