

# Advanced Data Set Composition with RNTuple



EP R&D

UNIVERSITY OF TWENTE.



Florine de Geus (University of Twente, CERN)

Vincenzo Eduardo Padulano (CERN)

Jakob Blomer (CERN)

Philippe Canal (FNAL)

Ana-Lucia Varbanescu (University of Twente)

**RNTuple** [1] is ROOT's next-generation columnar data format and I/O subsystem, aiming at:

- Less disk and CPU usage;
- Efficient use of modern hardware and object stores;
- Modern and robust interfaces.

To streamline analysis and reduce the need for storage resource, experiments may offer common, **centrally produced** data sets containing only a **fraction of all available physics objects**, while still covering **up to 80%** [2, 3] of the analysis use cases.

**Data set joins** open up the possibility for the remaining use cases to use custom event data **without having to duplicate** the common data.

Events between two data sets may be **out of order**. This can happen when data sets are produced and written in parallel.

CommonEvents		
Event	A	B
...		
1997		
1998		
1999		
...		

(primary)

joined with

CustomEvents	
Event	C
...	
1999	
1997	
1998	
...	

(secondary)

Event	A	B	C
...			
1997			
1998			
1999			
...			

Events may be **missing** from one data set. This can happen when (additional) filters have been applied.

CustomEvents	
Event	C
...	
1997	
1999	
...	

(secondary)

Event	A	B	C
...			
1997			
1998			-
1999			
...			

## Joins with RNTuple

Joining a **primary** and a **secondary** RNTuple requires the construction of an **index** for the secondary data set. This index maps one or more field values to entry numbers and is used to find entries corresponding to the ones in the primary data set.

**Questions to consider** include:

- How to handle one-to-many relations?
- What is the best representation for the index?
- Should the index be stored (and if so, how)?

## The RNTupleProcessor

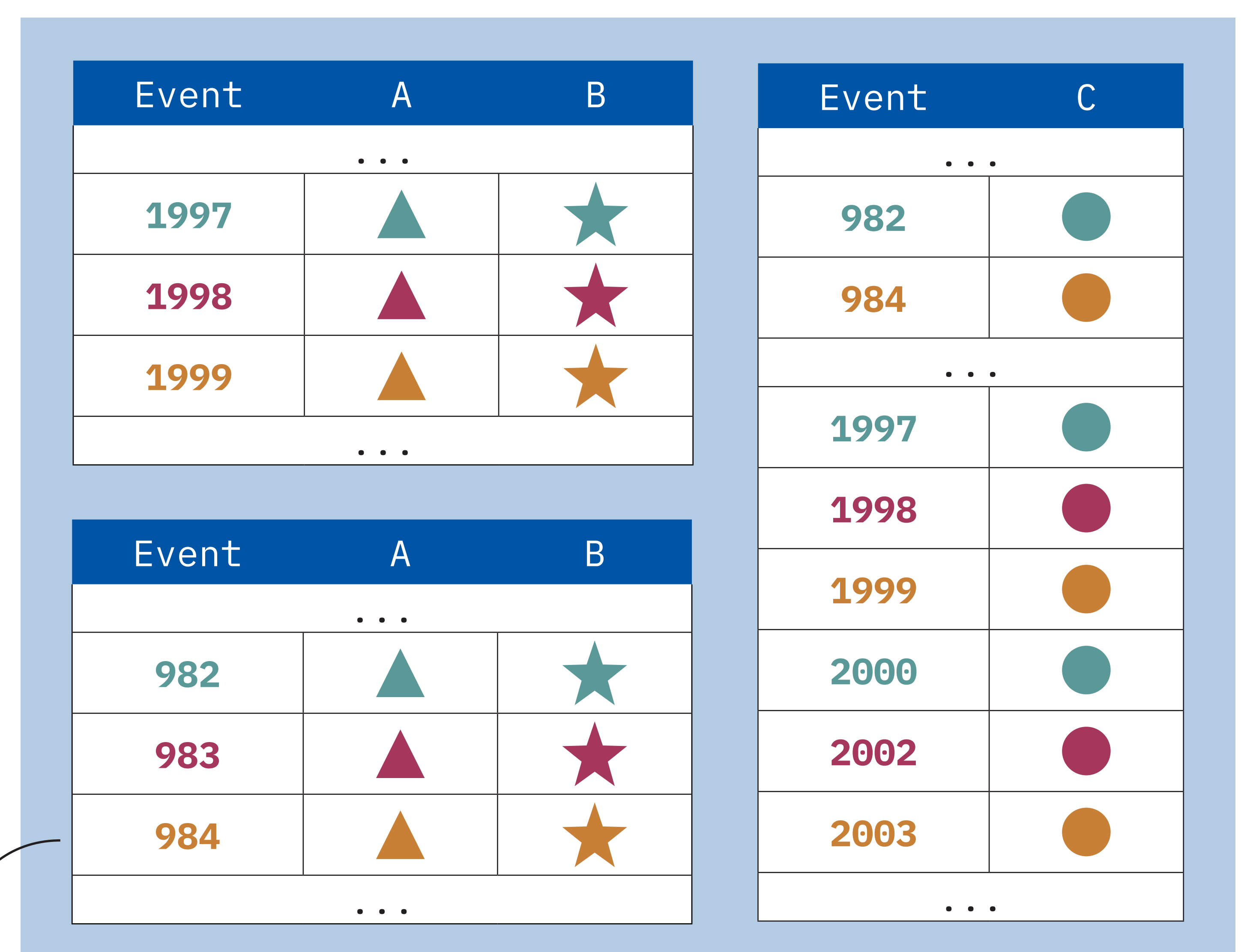
The **RNTupleProcessor** is a work in progress, aiming to:

- Provide a **low-level interface** to iterate over events;
- Enable **joins** and **vertical concatenations**;
- Serve as a backend for **RDataFrame** and other analysis frameworks.

A crucial design aspect are the **composition rules** that determine the order in which data sets are processed (i.e., horizontally, vertically or a combination). This will significantly influence **processing performance**.

Other **questions to consider** include:

- How to make the composition rules transparent to the user?
- How to deal with data distributed across multiple files?
- How to deal with distributed analysis?



Event	A	B	C
984			

The RNTupleProcessor provides an **iterator** over the composed data sets, representing a **view on the current entry**.

Contact and more information

florine.de.geus@cern.ch

<https://root.cern>

[1] J. Blomer, P. Canal, A. Naumann, and D. Piparo, "Evolution of the ROOT Tree I/O," EPJ Web Conf., vol. 245, 2020, doi: 10.1051/epjconf/202024502030.

[2] K. Ehatäht, "NANO AOD: a new compact event data format in CMS," EPJ Web Conf., vol. 245, p. 06002, 2020, doi: 10.1051/epjconf/202024506002.

[3] J. Schaarschmidt et al., "PHYSLITE - A new reduced common data format for ATLAS," EPJ Web of Conf., vol. 295, p. 06017, 2024, doi: 10.1051/epjconf/202429506017.