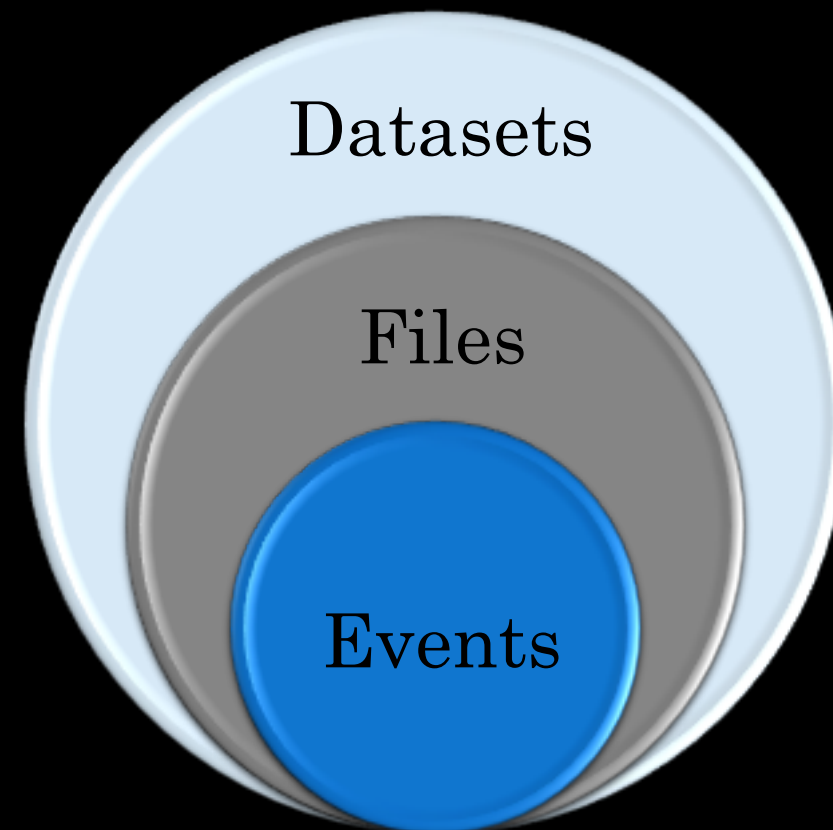
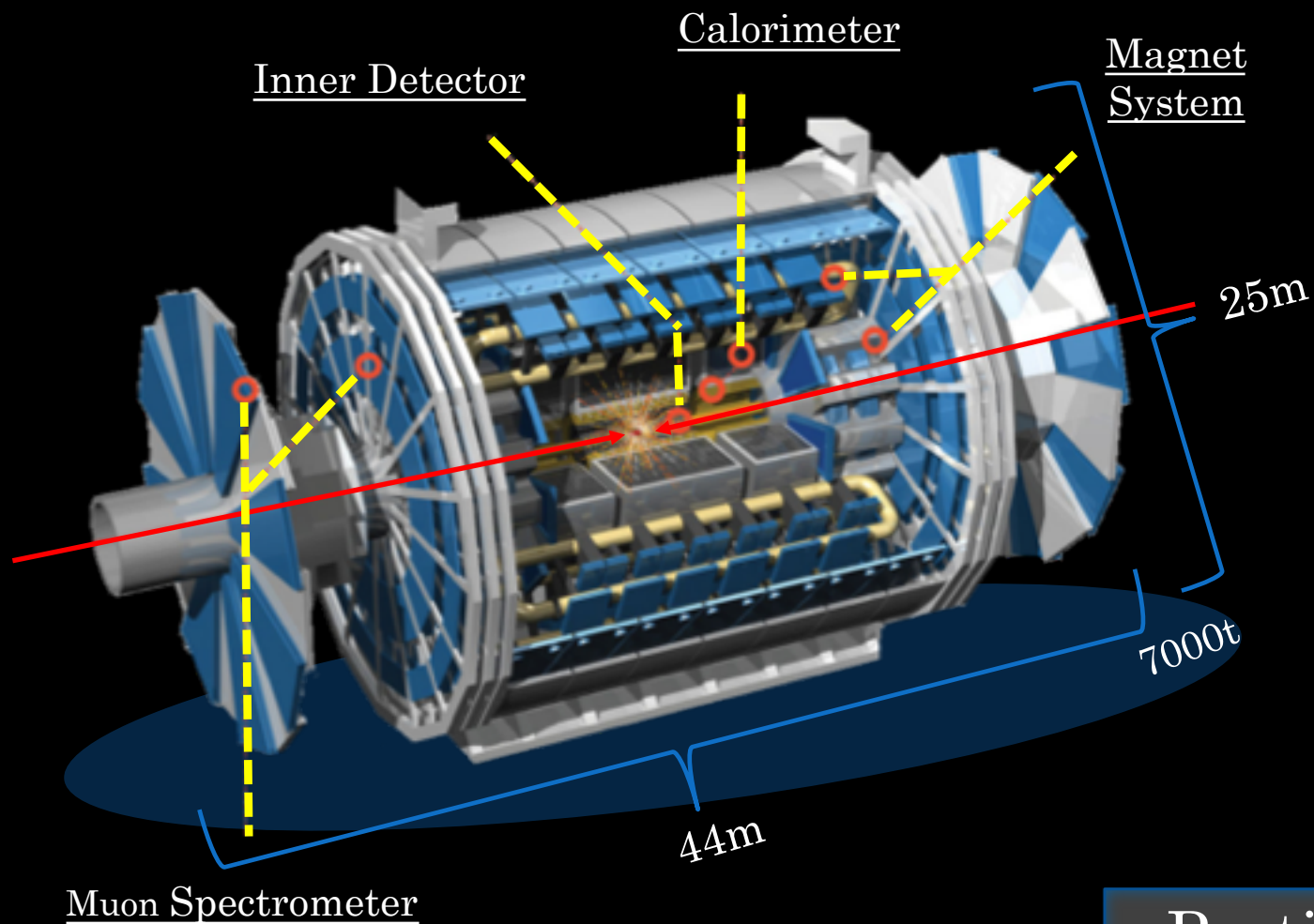


Events (particle collisions) in the ATLAS detector @LHC



Particle collisions are called **Events**



The one-million table partitions challenge in an ATLAS DB application

Gancho Dimitrov (CERN)



The Event WhiteBoard (EWB) project @ Oracle 18.3

- **EWB concept:** logically groups particle collision Events into collections
- **Collection:** events are processed in Event ranges
- **Collection removal:** once processing of a given collection is finished
- **Lifetime of an EWB collection:** from week(s) to month(s)

The EWB collections of data

TABLE object

COLL 20113	COLL 20114	COLL 22567	COLL 26101	COLL 27336
COLL 57310	COLL 58783	COLL 59933	COLL 63382	COLL 63388
COLL 63389	COLL 63870	COLL 64581	COLL 65002	COLL 65531
COLL 67763	COLL 67833	COLL 77345	COLL 77441	COLL 78803
COLL 78804	COLL 78818	COLL 79993	COLL 82233	COLL 82235
COLL 84433	COLL 86673	COLL 86674	COLL 86676	COLL 88332

→ TABLE partition

The EWB sponge in action

TABLE object

COLL 20113	COLL 20114	COLL 22567	COLL 26101	
COLL 57310	COLL 58783	COLL 59933	COLL 63382	
	COLL 63870		COLL 65002	COLL 65531
COLL 67763			COLL 77441	
COLL 78804	COLL 78818	COLL 79993	COLL 82233	
COLL 84433	COLL 86673	COLL 86674	COLL 86676	COLL 88332

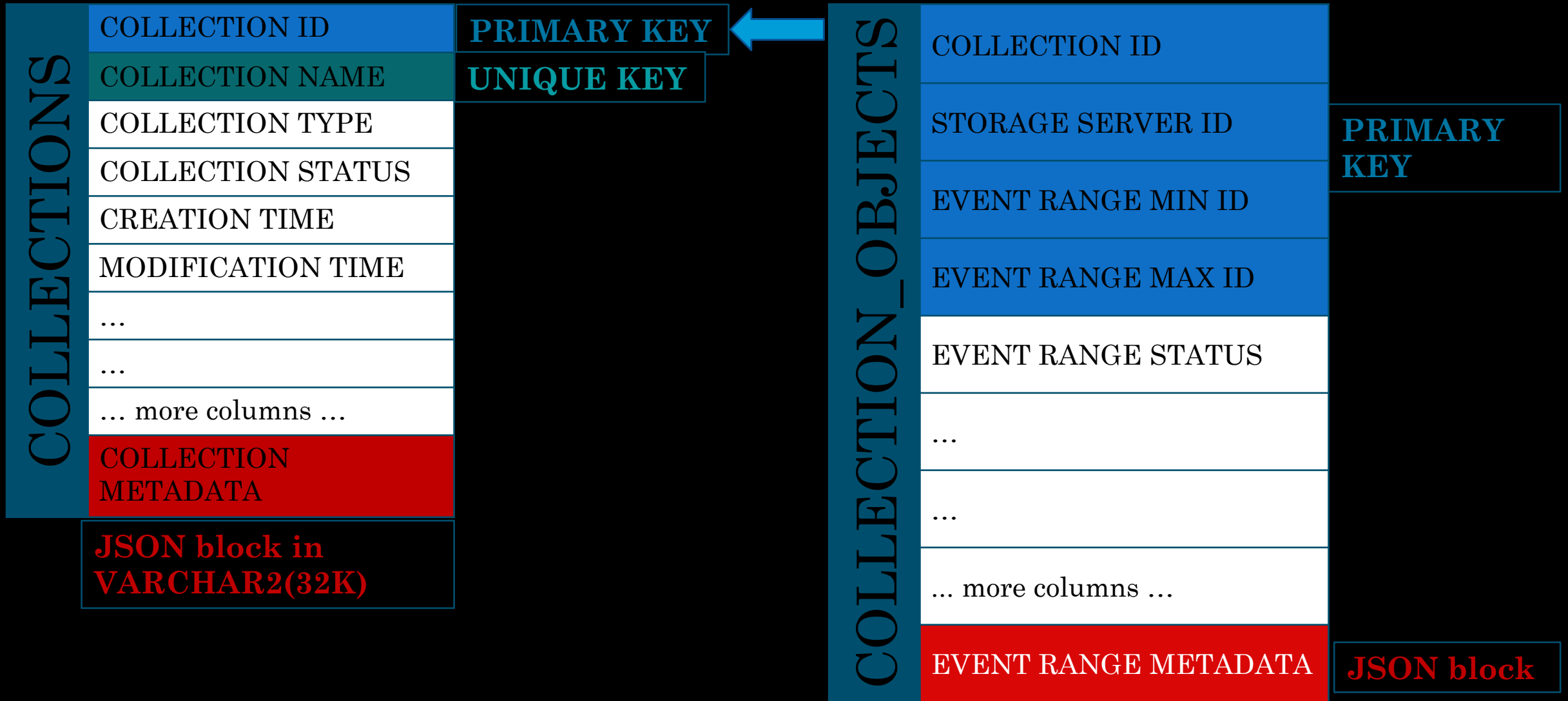
→ TABLE partition

Quiz

What is the maximum allowed number of partitions in a single table in Oracle RDBMS?

- 1) 100 thousand
- 2) 500 thousand
- 3) 1 million
- 4) More

EWB simplified tables layout (partial)



**How to partition the EWB
“collection_objects” child table?**

Design approach “1”
List partition per COLL_ID single value

Idea: each table partition contains data of a single EWB collection. **Removal of any EWB collection data would be straightforward.**

List-type partition for each data collection

```
CREATE TABLE COLLECTION_OBJECTS
(
  COLL_ID NUMBER(10,0),
  ...
  CONSTRAINT COLLOBJ_PK PRIMARY KEY (...) using index LOCAL
)
PARTITION BY LIST(COLL_ID) -- AUTOMATIC
( PARTITION COLLOBJ_ZERO VALUES(0) );
```

- Partition removal is easy:
`ALTER TABLE COLLECTION_OBJECTS DROP partition FOR (5276);`
- All worked well,
but does not seem scalable because of the 1048575 partitions limit per table
(error ORA-14299)

Design approach “2”
List partition per sequence of COLL_ID values

Idea: Each List-type table partition to host sequence of data collections (e.g. 10, 20, 50 or more collections per table partition).

Automation in List-type partitions creation (COLL_ID set)

- A dedicated List partition per set of collections is created by a **BEFORE INSERT** trigger on the parent table which calls an in-house created PLSQL procedure.



PARTITION_NAME	PART_POSITION	HIGH_VALUE
PART_COLL_251701	5036	251701, 251702, 251703, 251704, 251705, 251706, 251707, 251708, 251709, 251710, 251711, 251712, 251713, 251714, 251715
PART_COLL_251651	5035	251651, 251652, 251653, 251654, 251655, 251656, 251657, 251658, 251659, 251660, 251661, 251662, 251663, 251664, 251665
PART_COLL_251601	5034	251601, 251602, 251603, 251604, 251605, 251606, 251607, 251608, 251609, 251610, 251611, 251612, 251613, 251614, 251615
PART_COLL_251551	5033	251551, 251552, 251553, 251554, 251555, 251556, 251557, 251558, 251559, 251560, 251561, 251562, 251563, 251564, 251565
PART_COLL_251501	5032	251501, 251502, 251503, 251504, 251505, 251506, 251507, 251508, 251509, 251510, 251511, 251512, 251513, 251514, 251515
PART_COLL_251451	5031	251451, 251452, 251453, 251454, 251455, 251456, 251457, 251458, 251459, 251460, 251461, 251462, 251463, 251464, 251465
PART_COLL_251401	5030	251401, 251402, 251403, 251404, 251405, 251406, 251407, 251408, 251409, 251410, 251411, 251412, 251413, 251414, 251415
PART_COLL_251351	5029	251351, 251352, 251353, 251354, 251355, 251356, 251357, 251358, 251359, 251360, 251361, 251362, 251363, 251364, 251365

Interesting finding:

- **Achieved flexibility** as # sequential collections per partition can be changed by changing a single value in the "before insert" trigger:

Sequence of 10 collections: created 88485 partitions

Sequence of 5 collections: created 32745 partitions

- **After creation of 121230 partitions:**

"Error "ORA-14309: Total count of list values exceeds maximum allowed"

- What is the maximum number of list values in the Oracle DB ?

Count on the existing list partition key values showed :

1048575

Design approach “3”

List automatic partitions on virtual column based on COLL_ID

Idea: List partition on virtual column $\text{MOD}(\text{COLL_ID}, \text{nnn})$. It guarantees maximum “nnn” partitions on the child table (note: “nnn” must be smaller than 1 million)

Avoids the max partitions per table limit (**ORA-14299**) and the number of list-key values limit (**ORA-14309**).

List-type automatic table partitioning on virtual column

```
CREATE TABLE COLLECTION_OBJECTS
(
  COLL_ID NUMBER(10,0),
  COLL_ID_VIRT_GROUP NUMBER(10,0) GENERATED ALWAYS AS (MOD(COLL_ID,500000)) VIRTUAL,
  ...
  CONSTRAINT COLLOBJ_PK PRIMARY KEY (...) using index LOCAL
)
PARTITION BY LIST(COLL_ID_VIRT_GROUP)
AUTOMATIC
( PARTITION COLLOBJ_ZERO VALUES(0) );
```

MOD function returns the remainder of COLL_ID divided by 500000.

→ The table will have max 500K partitions

“List automatic” partitions on virtual column

- Test: 500000 partitions were automatically created using “INSERT INTO collection_objects ...” statement.
- It took about a week time.
Over the time, a partition creation was taking more time.

Upto 30K partitions: rate of 50-60 partitions/second

After 70K partitions: rate of 3-4 partitions/second

After 80K partitions: rate of 3 partitions/second

After 160K partitions: rate of 1-2 partitions/second

After 180K partitions: rate of 1 partition/second

After 200K partitions: rate of 1 partition/second

Within 200K-400K partitions: rate of 1 partition per 1-2 seconds

Within 400K-500K partitions: rate of 1 partition per 2 seconds

Best approach out of the explored five paths?

Detailed presentation (50min)

@BGOUG conference

7th-9th June

Borovets resort (Bulgaria)