Conquer Big Data with Oracle 18c, In-Memory External Tables and Analytic Functions

December 4, 2019

Jim Czuprynski

Senior Enterprise Data Architect

@JimTheWhyGuy

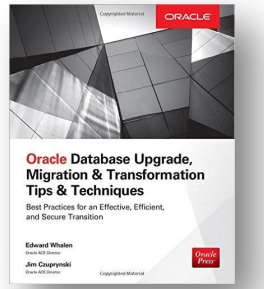
My Credentials

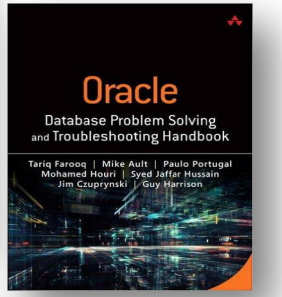
- Nearly 40 years of database-centric IT experience
- Oracle DBA since 2001
- Oracle 9i, 10g, 11g, 12c OCP, ADWC Certified
- Oracle ACE Director
- 100+ articles on databasejournal.com and IOUG SELECT
- Co-author of 4 Oracle books
- Oracle-centric blog (Generally, It Depends)
- Regular speaker at Oracle OpenWorld, COLLABORATE, KSCOPE, and Regional OUGs

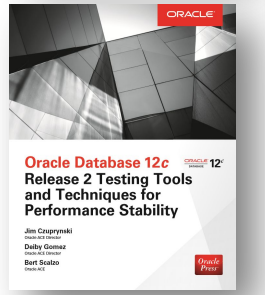
ORACLE Certified Professional

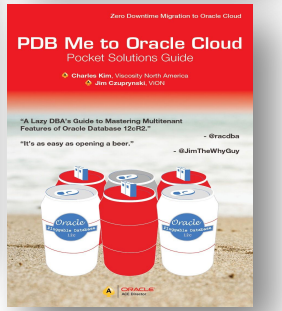
ORACLE ACE Director

ORACLE Certified Specialist









E-mail me at jczuprynski@zerodefectcomputing.com
Check out my **blog**: http://jimczuprynski.wordpress.com
Follow me on **Twitter: @JimTheWhyGuy**
Connect with me on LinkedIn: **Jim Czuprynski**

Our Agenda

- Big Data and IoT: Zettabytes, Here We Come!
- In-Memory External Tables: Crunching Through Data at Lightspeed
- Example: Using Oracle 18*c* to Analyze Credit Scoring Data
- New Analytic Functions: Close Enough For What We're Doing
- Oracle 19*c*: Hybrid Partitioned External Tables

Big Data and IoT: Zettabytes, Here We Come!

Big Data, the Internet of Things (IoT), and Analytics
have arrived for real … and human civilization
is already impacted tremendously

Some ***amazing statistics*** to ***consider*** …

- ***Big Data:*** **4.4** ZettaBytes in 2013, **44** ZettaBytes by 2020
- ***IoT:*** **25 billion** devices in 2015, **50 billion** by 2020
- **6.4 Billion** connections between IOT devices by 2025

… and be ***nervous about*****!**

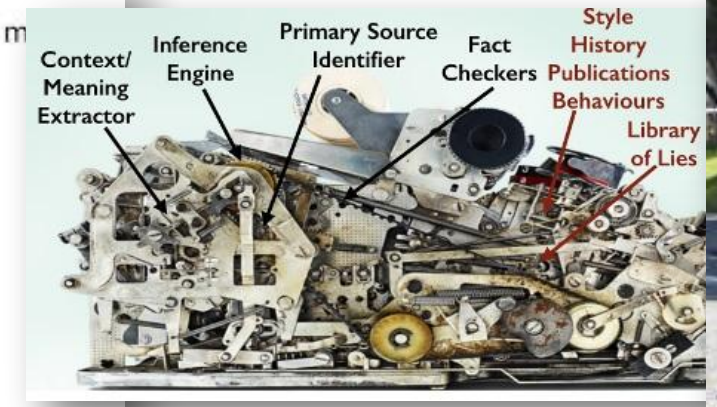
- Huge amount of sensitive data needs to be **secured** and **protected**
- Even though unstructured data abounds in data lakes, the biggest challenge is figuring out what data is **information** and which is **dreck**
- How do we tie existing datasets in RDBMS format to these new sources?

Big Data and Analytics: More Prevalent Than Ever

Big Data is now being analyzed in real time, and our civilization is never going to be the same again.

Predictive Analytics are *already* driving social change in real time

- ***Immigration:*** DHS contemplating use of ***credit scores*** to determine immigration cases
- ***Politics:*** Building a ***truth engine*** to assess veracity of spreading news stories
- ***Criminal Justice:*** Chicago police leveraging ***threat scores*** during traffic stops
- ***Civic Duty:*** China's new ***social credit*** system



The Sharper Knife You Already Own: The Case for SQL vs. NoSQL

Today, *NoSQL databases* are like *avocado toast* …

- **MongoDB** leverages key-value pairs stored within JSON documents
- **HDFS** (file system) and **HIVE** (database) use syntax ***similar*** to traditional Oracle SQL, but just different enough to introduce confusion
- Oracle even offers its own NoSQL database

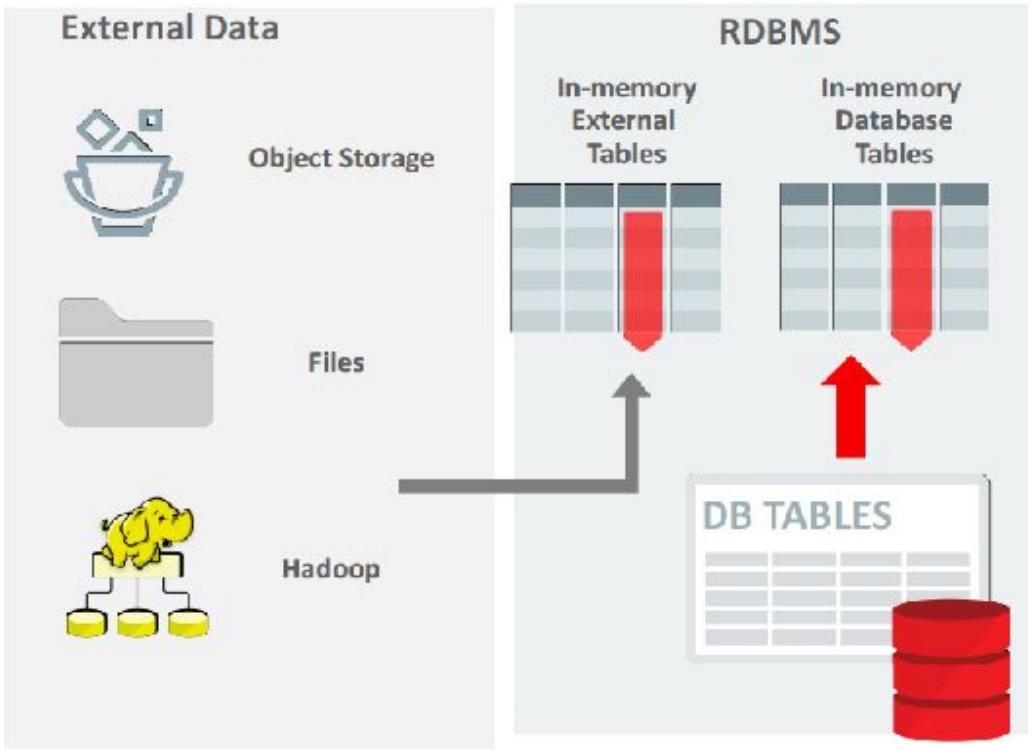
… but sometimes, you just gotta have *steak and eggs!*

- Oracle SQL is **already** considerably robust
- Oracle 12*c*R1: Read **directly** from JSON documents, HDFS files, and HIVE tables
- Oracle 12*c*R2: **Partitioned** external tables for faster processing
- Oracle 18*c:* **In-Memory** external tables

What if you could shorten your developers' learning curve … by *eliminating* it?

In-Memory External Tables: A Primer

In-Memory External Tables (IMXT)



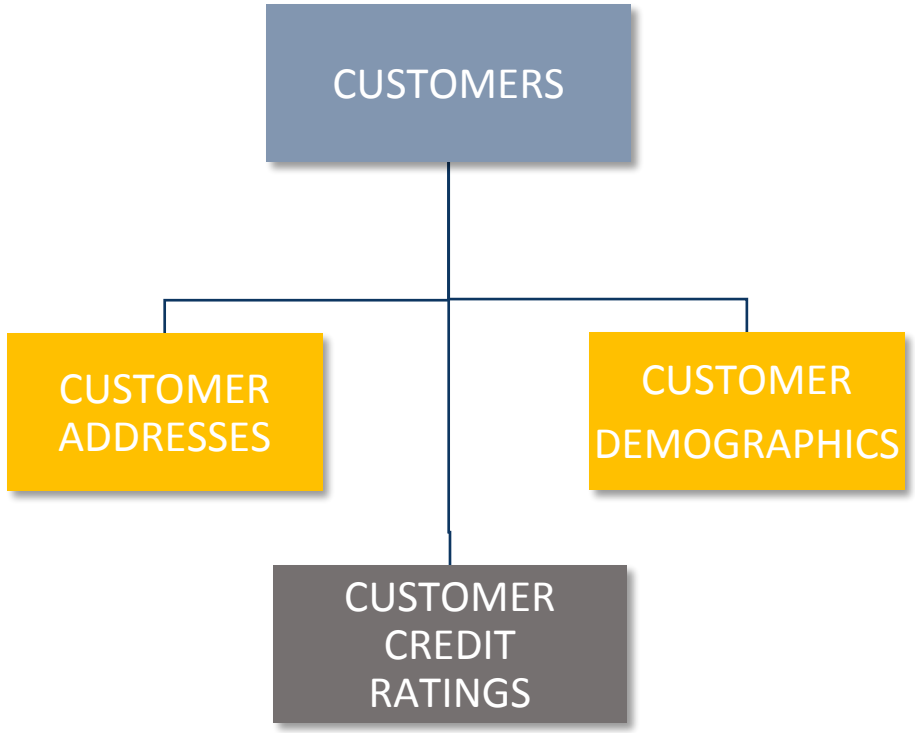
In-Memory External Tables (IMXT)

- Allows transparent access to data outside a traditional Oracle database
- Builds in-memory column cache of outside data
- Enables uber-fast analytics against external tables
- All in-memory optimizations apply, including vector processing and JSON expressions
- Tests promise potential performance improvement of as much as **100X**

In-Memory External Tables: Use Cases

Consider the data relationships between customers, address information, demographics, and credit ratings:

- Credit ratings encompasses a comparatively large amount of data that is probably undesirable to load into ORGANIZATION HEAP internal database tables
- Credit ratings are comparatively temporary data that's just needed for a brief analysis
- Ratings data spans several narrow but deep CSV-formatted files



Creating IMXTs

```
SQL> CREATE TABLE tpcds.imet_customer_credit_ratings (
      ccr_customer_number       NUMBER(8)
     ,ccr_last_reported         TIMESTAMP
     ,ccr_credit_rating         NUMBER
     ,ccr_missed_payments       NUMBER
     ,ccr_credit_maximum        NUMBER
)
  ORGANIZATION EXTERNAL (
    TYPE ORACLE_LOADER
    DEFAULT DIRECTORY imet_dir
    ACCESS PARAMETERS (
      RECORDS DELIMITED BY NEWLINE
      FIELDS TERMINATED BY ',' (
            ccr_customer_number      CHAR(07)
           ,ccr_last_reported        CHAR(20)
            DATE_FORMAT DATE MASK "DD-MON-YYYY HH24:MI:SS"
           ,ccr_credit_rating        CHAR(03)
           ,ccr_missed_payments      CHAR(01)
           ,ccr_credit_maximum       CHAR(06)
      )
  )
. . .
```

EXTERNAL TABLE driver

Directory on Oracle Database host

Row formatting parameters

```
. . .
LOCATION(
      'ccr_1998.csv'
     ,'ccr_1999.csv'
     ,'ccr_2000.csv'
     ,'ccr_2001.csv'
     ,'ccr_2002.csv'
   )
);

Table created.

BEGIN
  DBMS_STATS.GATHER_TABLE_STATS(
     ownname => 'TPCDS'
    ,tabname => 'imet_customer_credit_ratings'
    ,degree => 4);
END;
/
```

File names

IMXTs: Activation and Population

```
SQL> ALTER TABLE tpcds.imet_customer
  INMEMORY
    PRIORITY HIGH
    MEMCOMPRESS FOR QUERY LOW;

ALTER TABLE tpcds.imet_customer_addr
  INMEMORY
     PRIORITY HIGH
     MEMCOMPRESS FOR QUERY LOW;

ALTER TABLE tpcds.imet_customer_demo
  INMEMORY
     PRIORITY HIGH
     MEMCOMPRESS FOR QUERY LOW;

ALTER TABLE tpcds.imet_customer_credit_ratings
   INMEMORY
     MEMCOMPRESS FOR QUERY LOW;
```

ORGANIZATION HEAP tables

Needed for activities against IMXTs!

```
ALTER SESSION
  SET QUERY_REWRITE_INTEGRITY = stale_tolerated;

BEGIN
  DBMS_INMEMORY.POPULATE(
    schema_name => 'TPCDS'
   ,table_name => 'IMET_CUSTOMER_CREDIT_RATINGS'
  );
END;
/
```

Initiates IMXT population

ORGANIZATION EXTERNAL table.
Note absence of PRIORITY attribute!

IMXTs: Tracking IMCS Population and Usage

```
SQL> SELECT
       SS.name
      ,MS.value
  FROM
      v$sysstat SS
      ,v$mystat MS
  WHERE SS.statisti
    AND MS.value >
   AND (SS.name LIK
 ORDER BY SS.name;
```

How much IMCS-related activity is being generated while I'm running a query against IMXTs or normal tables in IMCS?

```
In-Memory Session-Level Statisti          V$MYSTAT)

                                                                Statistic
Statistic Name                                                      Value
---------------------------------------------------------------- ------------
IM scan CUs columns accessed                                             240
IM scan CUs columns theoretical max                                      525
IM scan CUs memcompress for query low                                    105
IM scan CUs pcode pred evaled                                            105
IM scan CUs split pieces                                                 112
IM scan bytes in-memory                                          484,622,076
IM scan bytes uncompressed                                       649,335,277
IM scan rows                                                      26,101,901
IM scan rows projected                                               220,905
IM scan rows valid                                                26,101,901
IM simd compare calls                                                    105
IM simd decode calls                                                     968
IM simd decode selective calls                                           968
```

IMXTs: Crunching Through Data at Lightspeed

```
SQL> ALTER SESSION SET QUERY_REWRITE_INTEGRITY = stale_tolerated;
```

```
SQL> ALTER SESSION SET inmemory_query = ENABLE;
```

```
SQL> SELECT
    ccr_missed_payments
   ,MIN(ccr_credit_rating)
   ,AVG(ccr_credit_maximu
   ,COUNT(*)
  FROM tpcds.imet_custo
 WHERE ccr_last_reporte
                                                    -01', 'yyyy-mm-dd')
                                                    -31', 'yyyy-mm-dd')
 GROUP BY ccr_missed
 ORDER BY ccr_miss
```

Leveraging the IMXT that's populated in IMCS improved performance by almost **20X**

N… …ssed in less than 51 seconds from flat files

```
Elapsed: 00:00:02.45

-----------------------------------------------------------------------------------------------------------------
| Id  | Operation                          | Name                        | Rows  | Bytes | Cost (%CPU)| Time     |
-----------------------------------------------------------------------------------------------------------------
|   0 | SELECT STATEMENT                   |                             |     5 |   115 | 24559   (2)| 00:00:01 |
|   1 |  SORT GROUP BY                     |                             |     5 |   115 | 24559   (2)| 00:00:01 |
|*  2 |   EXTERNAL TABLE ACCESS INMEMORY FULL| IMET_CUSTOMER_CREDIT_RATINGS |  5220K|   114M| 24393   (1)| 00:00:01 |
-----------------------------------------------------------------------------------------------------------------

Predicate Information (identified by operation id):
---------------------------------------------------

   2 - inmemory("CCR_LAST_REPORTED"<=TIMESTAMP' 2000-03-31 00:00:00' AND "CCR_LAST_REPORTED">=TIMESTAMP'
              1999-04-01 00:00:00')
       filter("CCR_LAST_REPORTED"<=TIMESTAMP' 2000-03-31 00:00:00' AND "CCR_LAST_REPORTED">=TIMESTAMP'
              1999-04-01 00:00:00')
```

IMXTs: Relevant Metadata

External Table Attributes	
View	**Description**
{USER | ALL |DBA}_TABLES	Lists tables, including EXTERNAL tables and their INMEMORY status
{USER | ALL |DBA}_TAB_COLUMNS	Lists table columns, including those for EXTERNAL tables
{USER | ALL |DBA}_EXTERNAL_TABLES	Lists all EXTERNAL tables
{USER | ALL |DBA}_EXTERNAL_LOCATIONS	Shows file locations for EXTERNAL tables
V$INMEMORY_AREA	Monitors utilization of IMCS space at SMU, IMCU, and IMEU levels, including progress of ongoing (re)population
V$INMEMORY_SEGMENT	Tracks which internal and external table segments are populated within the IMCS

New Analytic Capabilities:
Close Enough For What We're Doing

Top-N Approximate Aggregation Functions

Top-N Queries can leverage **APPROX_COUNT(), APPROX_SUM(), APPROX_RANK()**

```
SELECT
    ca_state
   ,ca_zip
   ,APPROX_COUNT(c_customer_sk) p
   ,APPROX_SUM(cd_dep_count) depe
   ,APPROX_RANK(
     PARTITION BY ca_state
     ORDER BY APPROX_COUNT(c_cust
     AS rnk_by_cust_count
   ,APPROX_RANK(
     PARTITION BY ca_state
     ORDER BY APPROX_SUM(cd_dep_c
     AS rnk_by_dep_count
  FROM
    tpcds.customer
   ,tpcds.customer_address
   ,tpcds.customer_demographics
```

```
. . .
 WHERE c_current_addr_sk = ca_address_sk
   AND c_current_cdemo_sk = cd_demo_sk
   AND ca_state IN ('IL','IN','WI')
   AND ca_zip IS NOT NULL
 GROUP BY
    ca_state
   ,ca_zip
 HAVING APPROX_RANK(
     PARTITION BY ca_state
     ORDER BY APPROX_COUNT(c_customer_sk) DESC) <= 5
    AND APPROX_RANK(
     PARTITION BY ca_state
     ORDER BY APPROX_SUM(cd_dep_count) DESC) <= 10
 ORDER BY
    ca_state;
```

Note that **both** of the windowing computations are usable as filters in the **HAVING** clause!

For more details on 18c Top-N APPROX functions, take a look at http://bit.ly/18c_APPROX

Analytic Views: Enhancements in 18*c*

Analytic Views (AVs) are fast becoming a crucial underpinning of Big Data SQL and real-time analytic technology.

Enhancements in Oracle 18*c* include:

- Support for Excel **MDX** format
- Ranking and statistical functions (**RANK_*, COVAR_*, STATS_*, PERCENTILE_***)
- Hierarchical expressions (**HIER_DEPTH, HIER_LEVEL, HIER_MEMBER_NAME**)
- Dynamic definition of calculations within SQL queries
- Broader support beyond star schemas (snowflake and flat denormalized fact tables)

Check out http://bit.ly/18c_AVs for information on 18c Analytic View enhancements

Analytic Views: FILTER FACT and ADD MEASURE

```
WITH av_ffam ANALYTIC VIEW AS (
  USING tpcds.av_mkt_geo_sales
  HIERARCHIES (avhy_dates, avhy_geography)
  FILTER FACT (
    avhy_dates
      TO level_name = 'QUARTER'

                      STRICT'
                    LIKE '%South%')

  ADD MEASURES (
    prior_period_qty AS
        (LAG(fct_qty)
          OVER (HIERARCHY avhy_dates OFFSET 1))
   ,prior_period_pctchg AS
        (LAG_DIFF_PERCENT(fct_qty)
          OVER (HIERARCHY avhy_dates OFFSET 1))
  )
)
. . .
```

FILTER FACT *removes* data from consideration *before* it's submitted for aggregation

```
. . .
SELECT
     avhy_dates.level_name time_level
    ,avhy_dates.member_name time_range
    ,avhy_geography.level_name geo_level
    ,avhy_geography.member_name geo_range
    ,fct_qty
    ,prior_period_qty
    ,prior_period_pctchg
  FROM av_ffam
  HIERARCHIES (avhy_dates
               ,avhy_geography)
 WHERE avhy_dates.level_name IN
('ALL','YEAR','QUA
   AND avhy_geog
 ORDER BY
       avhy_dates
      ,avhy_geography.hier_order
;
```

The **WITH** clause is now supported within a calling SQL statement

It's now possible to add new **MEASURE**s without having to recreate the entire AV

Oracle 19*c*: A Preview of Even More Power

- Partitioned External Tables (PETs)
- Hybrid Partitioned External Tables (HYPEs)

Partitioned External Tables in 12*c* & 18*c*

```
SQL> CREATE TABLE tpcds.xpet_customer_credit_ratings (
    ccr_customer_number      INTEGER
   ,ccr_last_reported        TIMESTAMP
   ,ccr_credit_rating        INTEGER
   ,ccr_missed_payments      INTEGER
   ,ccr_credit_maximum       INTEGER
)
  ORGANIZATIO
    TYPE ORAC
    DEFAULT D
    ACCESS PA
      RECORDS
      FIELDS
            C
           ,C
             DATE MASK "DD-MON-YYYY HH24:
           ,ccr_credit_rating        CHAR(
           ,ccr_missed_payments      C
           ,ccr_credit_maximu
      )
  )
)
. . .
```

```
. . .
PARTITION BY RANGE(ccr_last_reported) (
      PARTITION ccr_1998
        VALUES LESS THAN (TO_DATE('1999-01-01', 'yyyy-mm-dd'))
        LOCATION('ccr_1998.csv')
      PARTITION ccr_1999
                                                          yyy-mm-dd'))
      PARTITION ccr_2000
                                                          yyy-mm-dd'))
      PARTITION ccr_2001
                                                          yyy-mm-dd'))
      PARTITION ccr_2002
        VALUES LESS THAN (MAXVALUE)
        LOCATION('ccr_2002.csv')
```

```
SQL> ALTER TABLE tpcds.xpet_customer_credit_ratings
       INMEMORY
       MEMCOMPRESS FOR QUERY LOW;

ALTER TABLE tpcds.xpet_customer_credit_ratings INMEMORY
                                               *
ERROR at line 1:
ORA-30657: operation not supported on external organized table
```

Placing this table within the IMCS is not allowed until **Oracle 19*c***.

Available in 19*c*: Hybrid Partitioned Tables (HyPTs)

NEW IN 19c

For the first time, we'll be able to describe a relationship between **internal** and **external** partitions as if it were a single table

```
CREATE TABLE tpcds.hypt_customer_credit_ratings (
  ccr_customer_number     INTEGER
 ,ccr_last_repor
 ,ccr_credit_rat
 ,ccr_missed_pay
 ,ccr_credit_max
PARTITION BY RAN
    PARTITION cc
      VALUES LES
      TABLESPACE
   . . .
   ,PARTITION cc
      VALUES LES
    TABLESPACE t
```

```
ALTER TABLE tpcds.hypt_customer_credit_ratings
  ADD EXTERNAL PARTITION
    ATTRIBUTES (
      TYPE ORACLE_LOADER
      DEFAULT DIRECTORY tpcds_hypt
      ACCESS PARAMETERS (
         FIELD
          ccr_
         ,ccr_
            DA
         ,ccr_
         ,ccr_
         ,ccr_
```

```
ALTER TABLE tpcds.hypt_customer_credit_ratings
  ADD PARTITION ccr_2017
      VALUES LESS THAN (TO_DATE('2018-01-01', 'yyyy-mm-dd'))
      EXTERNAL LOCATION ('CCR_2017.csv');

ALTER TABLE tpcds.hypt_customer_credit_ratings
  ADD PARTITION ccr_2018
      VALUES LESS THAN (MAXVALUE)
      EXTERNAL LOCATION ('CCR_2018.csv');
```

Oracle 19*c*: Expanded Support for IMXT

NEW IN 19c

Big Data and Performance Enhancements for In-Memory External Tables

- In-Memory External Tables add support for **ORACLE_HIVE** and **ORACLE_BIGDATA** drivers, parallel query, Oracle Real Application Clusters, Oracle Active Data Guard, and **on-demand** population.
- By using the new Big Data drivers, you avoid the cost and complexity of materializing data **before** populating it into the In-Memory Column Store (IMCS).
- You can use the SQL analytical capabilities of Oracle Database and Database In-Memory to analyze **both internal and external data**.
- Support for parallel query and full scan population means **applications have fewer limitations** when accessing data that reside **outside** the database.

IMXTs: 18*c* Limitations vs. 19*c* Enhancements

IMXT / DBIM Feature	18*c*	19*c*
Are IMXTs populated / repopulated automatically via Automatic In Memory?	No	Yes
Is the PRIORITY clause supported for IMXTs?	No	No
Can I place specific COLUMNs from an IMXT into IMCS?	No	No
Is the DISTRIBUTE directive permitted?	No	Yes
Can the external files reside in HDFS?	No	Yes
Can I partition an IMXT?	No	Yes
Are Join Groups permitted?	No	No
Are In-Memory Expressions supported?	No	No
Is Parallel Execution supported?	No	Yes
Will In-Memory Optimized Arithmetic work for numeric column values?	No	No

References and Resources

Leverage these great resources to increase understanding of and enable experiments in IMXTs, APPROX functions, and Analytic Views for Oracle 18*c*, 19*c*, and beyond:

- **DataGenerator:** https://dominicgiles.com/datagenerator
- **IMXT 18*c* Documentation:** http://bit.ly/18c_IMETs
- **IMXT 19*c* Documentation:** http://bit.ly/19c_IMETs
- **18*c* APPROX Function Improvements:** http://bit.ly/18c_APPROX
- **18*c* Analytic View Improvements:** http://bit.ly/18c_AVs