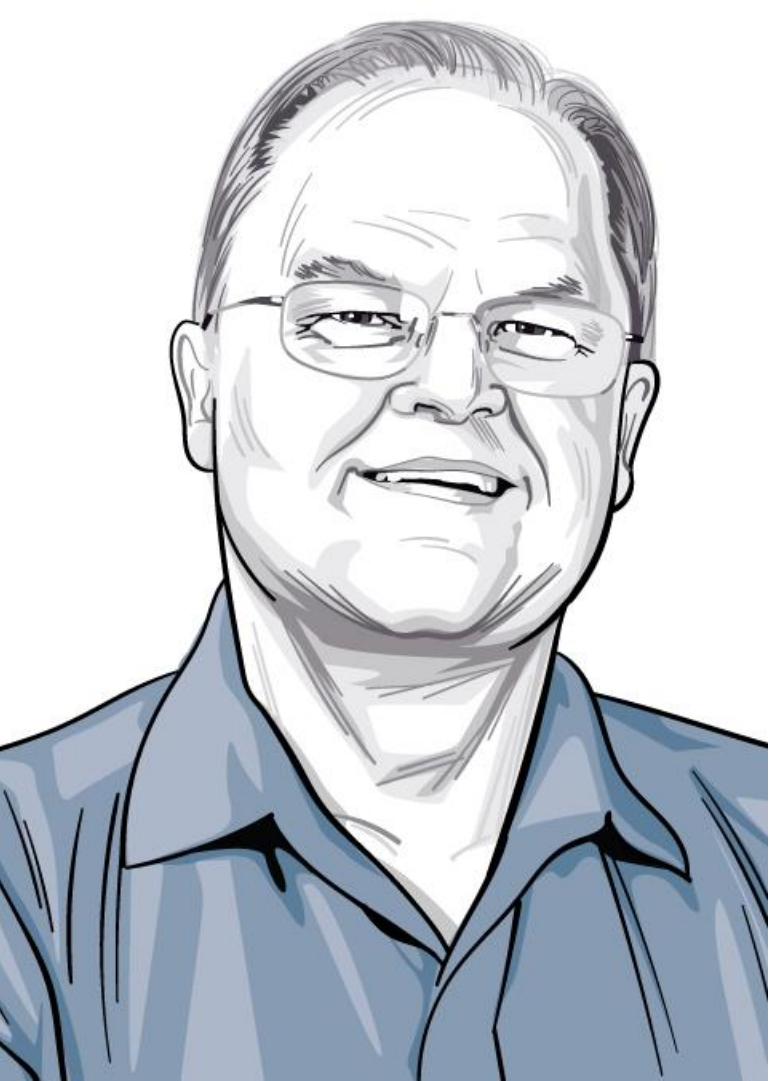
Oracle Extended Data Types

Just Don't……..

Quest®

Where Next Meets Now.



About Me

Oracle since Version 6

Oracle 7 Alpha

Oracle Very Large Database Focus Group

MOSES (Massive Open Systems Environment Specifications)

Oak Table

10 years in Search & Rescue/Emergency Management

Quest
Where Next Meets Now.

Agenda

- What are Extended Data Types?
- Why use Extended Data Types
- Side Trip – Defensive Software
- Why NOT use Extended Data Types
- Alternatives to Extended Data Types
- Q & A

What are Extended Data Types?

Large Strings

- **Extends VARCHAR2 and NVARCHAR2 from 4000 to 32767 bytes**
- **Extends RAW from 2000 to 32767 bytes**
- **Uses MAX_STRING_SIZE initialization parameter**
  - MAX_STRING_SIZE = STANDARD default
  - MAX_STRING_SIZE = EXTENDED large limits

Large Strings - Requirements

- **STANDARD to EXTENDED ONLY**
  - No going back
- **REQUIRES utl32k and utlrp scripts**
- **Special steps for Container and Pluggable Databases**
- **Special steps for RAC and Data Guard**

Large Strings – Side Effects

- **VARCHAR2 and NVARCHAR2 stored out-of-line**
  - But you can't use DBMS_LOB
- **RAW may be stored out-of-line if bigger than 4000 bytes**
- **Indexes are limited**
  - "It depends"
  - "approximately 6400 bytes"
  - There are workarounds
- **Partition Key list of values limited to 4K bytes**
- **Concatenation total length is 32,767 bytes**
- **NLS Implications - see Oracle Documentation**
- **XML Implications – see Oracle Documentation**

Why use Extended Data Types

Why use Extended Data Types?

- **You need large strings**
  - Data from a legacy non-Oracle system
  - External data
  - Need to use Oracle string functions
  - Large strings that cannot use CLOBs

Side trip – Defensive Software

Defensive Software

- **Wikipedia**
  - Defensive programming is an approach to improve software and source code, in terms of:
    - General quality – reducing the number of software bugs and problems.
    - Making the source code comprehensible – the source code should be readable and understandable so it is approved in a code audit.
    - Making the software behave in a predictable manner despite unexpected inputs or user actions.
      A must for mission-critical software
- **What's mission-critical?**
  - Medical Devices
  - Critical Infrastructure
  - Databases?
  - SQL?

Databases and SQL ARE Mission Critical

- **What does downtime cost for YOUR database?**
  - eCommerce
  - Manufacturing
    - Process restart
    - Missed deadlines
  - HR
  - CRM
    - Lost sales
  - Financials
    - Period Close
    - Payroll
  - Test/Dev
    - How many developers sitting on their hands?

Why NOT use Extended Data Types

Why NOT use Extended Data Types?

- You don’t need large strings
- Defensive Programming/Operations

High level concerns

- **It's irreversible**
- **It requires an outage or migration**
- **It requires execution of additional scripts**
  - Utl32k
  - utlrp
- **MultiTenant, RAC and DataGuard require special handling**
- **Globalization requires special handling**
- **Dev/Test/QA will all require changes**
- **Extended Data Type columns are stored out-of-line**
  - Performance issues
  - Without ASM – BasicFile LOBs – may be deprecated

Application Concerns

- **ALL code that works with strings is at risk**
  - Is ALL your code protected against string overflow?
  - Possible exploit vectors
- **Indexes are at risk**
  - Index key length limits
  - could cause run-time errors
- **Partitions are at risk**
  - List partition limit
  - Could cause run-time errors
- **XML types are at risk**
  - Expression size limit

String Handling Concerns

- **Utl32k script will update columns for views “where this is required”**
- **Utl32k “does not increase…some views because of the way the SQL for these views is written”**
- **Application code needs to be updated**
  - Failure to update can cause run-time errors

Performance concerns

- Out of line LOB handling
- Extra storage space
- More data blocks

Alternatives to Extended Data Types

Alternatives

- **Just don’t….**
  - Do you REALLY need character data types that large?
- **Can you split the data?**
  - Multiple smaller columns
- **CLOBs**
  - Character LOBs

LOBS

- **Tried and true**
- **Requires explicit extra work**
  - DBMS_LOB
- **Some restrictions**
  - Can’t be a key
  - No Clusters
  - No sorts
  - No joins
  - See Oracle Documentation

Conclusions

Conclusions

- **It depends**
- **There are tradeoffs**
  - Pros
    - Easier string handling
    - Eases SOME LOB limitations
  - Cons
    - Still really LOBs
    - Different limitations
- **Be defensive!**
- **Know your data!**

Questions?

Quest
Where Next Meets Now.


Thank You
Quest
Where Next Meets Now.