

## 5 Data Warehousing/Big Data

This section describes the new data warehousing/big data features.

### Enhanced Partitioning Metadata

Data dictionary views that contain partitioning-related metadata, for example, `ALL_TAB_PARTITIONS`, have two additional columns representing the high value (boundary) information of partitions and subpartitions in JSON and CLOB format.

Providing the high value (boundary) partitioning information in JSON and as CLOB allows you to use this information programmatically. This enables simple and automated processing of this information for schema retrieval or lifecycle management operations.

[View Documentation](#)

### Extended Language Support in Oracle Text

Language support is extended in Oracle Text, now supporting up to 48 languages. Additionally, there is extended support for all languages. To avoid the extended language support increasing your install footprint on disk, a new mechanism is introduced to control the downloaded languages on demand.

Customers benefit from the improved support for languages and can download the required files for only the languages they support. They can avoid using disk space unnecessarily for unneeded languages.

[View Documentation](#)

### External Table Partition Values in the File Path

External tables pointing to data in the object store can consist of a large number of files. These files can be organized across multiple directories, and even directory trees. You can use external table partitioning with folder names as part of the file paths. External table columns can also now return the file name of the source file for each row. The partition values can be derived from the directory name or file name.

External tables and SQL\*Loader can load large numbers of data files in object stores and meet the requirements for Hive-generated tables organized across multiple directories, and even directory trees. This feature enables external table

partitioning based upon the directory and file name; for example, you can have files for different months or for different states in separate directories.

[View Documentation](#)

## **Logical Partition Change Tracking for Materialized View Refresh and Staleness Tracking**

Logical Partition Change Tracking (LPCT) tracks the staleness of materialized views. LPCT operates at a fine level of logical granularity and gives you the flexibility to align the boundaries of logical partitions with the business rules and with changes applied to tables. It evaluates the staleness of the base tables for individual logical partitions without using a materialized view log or requiring any of the tables used in the materialized view to be partitioned.

With LPCT, materialized view staleness is tracked at the granularity of the logical partitions. This functionality significantly broadens the applicability of query rewrite for your application due to the fine-grained query rewrite. With LPCT, you perform refresh operations targeted at stale logical partitions only, improving the refresh time and avoiding complete re-loading data.

[View Documentation](#)

## **Staging Tables**

Staging tables are heap tables optimized for fast data ingestion and for handling volatile data. Key table attributes are set to defaults for these use cases without any additional user interaction.

Creating staging tables rather than 'normal' tables saves you time and effort so that you do not need to tune your table attributes for fast data ingestion with volatile data content. A staging table is configured by default with optimal configuration settings in order to guarantee the best possible performance and to avoid unnecessary performance debugging and tuning.

[View Documentation](#)