DZ_SPIDX

- Build ID: 4
- TFS Change Set: 8291

Utilities for the management of Oracle MDSYS.SPATIAL_INDEX domain indexes

Summary

DZ_SPIDX

Functions

dz_spidz_main.geodetic_XY_diminfo	Function to quickly return a “default” geodetic dimensional info array.
dz_spidz_main.geodetic_XYZ_diminfo	Function to quickly return a “default” 3D geodetic dimensional info array.
dz_spidz_main.geodetic_XYM_diminfo	Function to quickly return a “default” LRS geodetic dimensional info array.
dz_spidz_main.geodetic_XYZM_diminfo	Function to quickly return a “default” 3D LRS geodetic dimensional info array.
dz_spidz_main.get_spatial_indexes	Function to harvest into list of dz_spidx objects all spatial indexes on a given table.
dz_spidz_main.flush_spatial_indexes	Function to harvest into list of dz_spidx objects all spatial indexes on a given table and subsequently drop those indexes.
dz_spidz_main.recreate_spatial_indexes	Procedure to recreate all spatial indexes documented in the collection of dz_spidx objects.
dz_spidz_main.recreate_spatial_indexes	Rebuilding spatial domain indexes using index rebuild DDL may be problematic for a number of reasons.
dz_spidz_main.spatial_mview_refresh	Refreshing an Oracle materialized view with a spatial domain index may generate punishing performance problems.
dz_spidz_main.sdo_join_check	Utilizing SDO_JOIN in cross-schema fashion is highly problematic as often the outsider schema lacks privledges on the domain index tables needed to utilize SDO_JOIN.
dz_spidz_main.sdo_join_check_verbose	Utilizing SDO_JOIN in cross-schema fashion is highly problematic as often the outsider schema lacks privledges on the domain index tables needed to utilize SDO_JOIN.

Functions

dz_spidz_main.geodetic_XY_diminfo

Function to quickly return a “default” geodetic dimensional info array.

Parameters

None

Returns

MDSYS.SDO_DIM_ARRAY collection

Notes

- Assumes 5 centimeter tolerance for all geodetic spatial information.

dz_spidz_main.geodetic_XYZ_diminfo

Function to quickly return a “default” 3D geodetic dimensional info array.

Parameters

p_z_lower_bound	optional override for lower Z bound (default -15000)
p_z_upper_bound	optional override for upper Z bound (default 15000)
p_z_tolerance	optional override for Z tolerance (default 0.001 units)

Returns

MDSYS.SDO_DIM_ARRAY collection

Notes

- Assumes 5 centimeter tolerance for all geodetic spatial information.

dz_spidz_main.geodetic_XYM_diminfo

Function to quickly return a “default” LRS geodetic dimensional info array.

Parameters

p_m_lower_bound	optional override for lower M bound (default 0)
p_m_upper_bound	optional override for upper M bound (default 100)
p_m_tolerance	optional override for M tolerance (default 0.00001 units)

Returns

MDSYS.SDO_DIM_ARRAY collection

Notes

- Assumes 5 centimeter tolerance for all geodetic spatial information.
- M defaults represent common reach measure system used in the US National hydrology dataset.

dz_spidz_main.geodetic_XYZM_diminfo

Function to quickly return a “default” 3D LRS geodetic dimensional info array.

Parameters

p_z_lower_bound	optional override for lower Z bound (default -15000)
p_z_upper_bound	optional override for upper Z bound (default 15000)
p_z_tolerance	optional override for Z tolerance (default 0.001 units)
p_m_lower_bound	optional override for lower M bound (default 0)
p_m_upper_bound	optional override for upper M bound (default 100)
p_m_tolerance	optional override for M tolerance (default 0.00001 units)

Returns

MDSYS.SDO_DIM_ARRAY collection

Notes

- Assumes 5 centimeter tolerance for all geodetic spatial information.
- M defaults represent common reach measure system used in the US National hydrology dataset.

dz_spidz_main.get_spatial_indexes

Function to harvest into list of dz_spidx objects all spatial indexes on a given table.

Parameters

p_owner optional owner name of table to be inspected p_table_name table to be inspects for spatial indexes

Returns

dz_spidx_list collection

Notes

- The list of dz_spidx will have a count of zero if no spatial indexes are discovered.

dz_spidz_main.flush_spatial_indexes

Function to harvest into list of dz_spidx objects all spatial indexes on a given table and subsequently drop those indexes.

Parameters

p_owner optional owner name of table to be inspected p_table_name table to be inspects for spatial indexes

Returns

dz_spidx_list collection

Notes

- Obviously the user must have permission to drop the indexes for this function to succeed.
- The list of dz_spidx will have a count of zero if no spatial indexes are discovered.

dz_spidz_main.recreate_spatial_indexes

Procedure to recreate all spatial indexes documented in the collection of dz_spidx objects.

Parameters

p_index_array	dz_spidx_list collection of dz_spidx objects

Returns

Nothing

Notes

- Obviously the user must have permission to create the indexes for this function to succeed.

dz_spidz_main.recreate_spatial_indexes

Rebuilding spatial domain indexes using index rebuild DDL may be problematic for a number of reasons. For example an online rebuild will require the spatial index exist twice on disk until the final swap removes the old version. This can create storage management problems for very large indexes. Often the simple solution is to just drop and recreate the index. This procedure wraps together the step for this task using dz_spidx to persist the details of the spatial index so you do not have to.

Parameters

p_filter	use to limit the spatial rebuilds to a given set of tables. The filter is simply table names LIKE '%' || p_filter || '%'
p_tablespace	optional parameter to change the domain index tablespace used.
p_quiet	optional TRUE or FALSE parameter to log details of rebuild action to

DBMS_OUTPUT.

Returns

Nothing

Notes

- Note that details of the spatial index are not stored anywhere permanently during the rebuild process. If for some reason your rebuild fails (space issues perhaps), the details of the spatial index are lost and you will need to recreate the index from your own DDL documentation.

dz_spidz_main.spatial_mview_refresh

Refreshing an Oracle materialized view with a spatial domain index may generate punishing performance problems. Usually there is little to be done other than drop the spatial index, refresh the materialized view and then recreate the index afterwards. The following procedure inspects a given materialized view, collects information on the spatial indexes, drop those spatial indexes, executes the refresh and then replaces the spatial indexes.

Parameters

list	materialized view refresh parameters
method	materialized view refresh parameters
rollback_seg	materialized view refresh parameters
push_deferred_rpc	materialized view refresh parameters
refresh_after_errors	materialized view refresh parameters
purge_option	materialized view refresh parameters
parallelism	materialized view refresh parameters
heap_size	materialized view refresh parameters
atomic_refresh	materialized view refresh parameters
nested	materialized view refresh parameters

Returns

Nothing

Notes

- For information on the procedure parameters see Oracle documentation on DBMS_MVIEW.REFRESH.
- DZ_SPIDX currently has no functionality to persist the details of a given spatial index outside the scope of it's current process. If your materialized view refresh crashes for some reason, the information about the dropped spatial index is lost and will need to be recreated from your DDL documentation.

dz_spidz_main.sdo_join_check

Utilizing SDO_JOIN in cross-schema fashion is highly problematic as often the outsider schema lacks privledges on the domain index tables needed to utilize SDO_JOIN. Even when the permissions are granted, the next time the spatial index is rebuilt the problem will reoccur. Similarly a missing spatial index will equally hose the spatial join. This function return TRUE or FALSE regarding whether SDO_JOIN is currently possible between two tables.

Parameters

p_table_name1	[owner.]table_name of first table in join
p_column_name1	column name of first table in join

p_table_name2	[owner.]table_name of second table in join
p_column_name2	column name of second table in join

Returns

VARCHAR2 text of either TRUE or FALSE

Notes

- For information on the actual problem use the sdo_join_check_verbose version.

dz_spidz_main.sdo_join_check_verbose

Utilizing SDO_JOIN in cross-schema fashion is highly problematic as often the outsider schema lacks privledges on the domain index tables needed to utilize SDO_JOIN. Even when the permissions are granted, the next time the spatial index is rebuilt the problem will reoccur. Similarly a missing spatial index will equally hose the spatial join. This function return details on what actions or permissions are needed in order to execute a spatial join between two tables.

Parameters

p_table_name1	[owner.]table_name of first table in join
p_column_name1	column name of first table in join
p_table_name2	[owner.]table_name of second table in join
p_column_name2	column name of second table in join

Returns

VARCHAR2 text or either TRUE or an explanation of the current problem

Notes

- This functions assumes the basics that you can see the tables in question and thus interrogate table metadata to discover the names of the domain index tables. The main results will be the exact domain table name that you need granted select permission upon to accomplish the spatial join.