Oracle Spatial FOURSQUARE: Table Management

AskTom 30. Sep. 2025

Create Credentials

```
BEGIN
DBMS_CLOUD.CREATE_CREDENTIAL(
  credential_name => ''OCI_CRED',
  username => 'OracleIdentityCloudService/me@example.com',
  password => '..Auth Tokens..' );
END;
```

List some files ("places") from object storage

```
SELECT object_name, bytes, last_modified FROM
DBMS_CLOUD.LIST_OBJECTS('OCI_CRED','https://objectstorage.eu-frankfurt-1.oraclecloud.com/n/..../o/') where object_name like 'places-0000%';
```

Create external table for categories

```
BEGIN
DBMS_CLOUD.CREATE_EXTERNAL_TABLE(
  table_name =>'foursquare_categories_ext',
   credential_name =>'OCI_CRED',
   file_uri_list =>'https://objectstorage.eu-frankfurt-1.oraclecloud.com/n/.../o/categories.zstd.parquet',
   format => '{"type": "parquet", "schema": "first"}' );
END;
```

```
%sql
SELECT count(*) FROM foursquare_categories_ext;
```

```
COUNT(*)
1245
```

Create an external table with all columns in the parquet file

```
BEGIN
DBMS_CLOUD.CREATE_EXTERNAL_TABLE(
   table_name =>'foursquare_desc_ext',
   credential_name =>'OCI_CRED',
   file_uri_list =>'https://objectstorage.eu-frankfurt-1.oraclecloud.com/n/.../o/places-00000.zstd.parquet',
   format => '{"type": "parquet", "schema": "first"}' );
END;
```

Create table based on external table

```
CREATE TABLE foursquare_desc AS
SELECT * FROM foursquare_desc_ext;
```

Describe table

```
%script
DESCRIBE  foursquare_desc;
```

```
Name              Null?    Type
FSQ_PLACE_ID               VARCHAR2(32767)
NAME                       VARCHAR2(32767)
LATITUDE                   BINARY_DOUBLE
LONGITUDE                  BINARY_DOUBLE
ADDRESS                    VARCHAR2(32767)
LOCALITY                   VARCHAR2(32767)
REGION                     VARCHAR2(32767)
POSTCODE                   VARCHAR2(32767)
ADMIN_REGION               VARCHAR2(32767)
POST_TOWN                  VARCHAR2(32767)
PO_BOX                     VARCHAR2(32767)
COUNTRY                    VARCHAR2(32767)
DATE_CREATED               VARCHAR2(32767)
DATE_REFRESHED             VARCHAR2(32767)

Name                  Null?    Type
DATE_CLOSED                    VARCHAR2(32767)
TEL                            VARCHAR2(32767)
WEBSITE                        VARCHAR2(32767)
EMAIL                          VARCHAR2(32767)
FACEBOOK_ID                    NUMBER(19)
INSTAGRAM                      VARCHAR2(32767)
TWITTER                        VARCHAR2(32767)
FSQ_CATEGORY_IDS               JSON
FSQ_CATEGORY_LABELS            JSON
PLACEMAKER_URL                 VARCHAR2(32767)
UNRESOLVED_FLAGS               JSON
GEOM                           BLOB
BBOX                           JSON

--------------------------
```

Create an external table for some columns in the parquet file

```
BEGIN
DBMS_CLOUD.CREATE_EXTERNAL_TABLE(
  table_name => 'foursquare_ext',
   credential_name => 'OCI_CRED',
   file_uri_list => 'https://objectstorage.eu-frankfurt-1.oraclecloud.com/n/../o/places-
*.zstd.parquet',
   format => json_object('type' value 'parquet'),
   column_list => 'FSQ_PLACE_ID VARCHAR2(4000), NAME VARCHAR2(4000) ,
LATITUDE NUMBER, LONGITUDE NUMBER, ADDRESS VARCHAR2(4000),
   LOCALITY VARCHAR2(4000),REGION VARCHAR2(4000),POSTCODE
VARCHAR2(4000), COUNTRY VARCHAR2(4000), FSQ_CATEGORY_IDS JSON' );
END;
```

Count number of rows in the table

```
%sql
SELECT count(*) FROM foursquare_ext;
```

```
COUNT(*)
105292526
```

Count number of distinct IDs in the table

```
%sql
SELECT count(distinct fsg_place_id) FROM foursquare_ext;
```

```
COUNT(DISTINCTFSQ_PLACE_ID)
105292526
```

Count number of rows/points for each country (sort and use country for table partitioning)

```
%sql
SELECT country , count(*) as cnt_points
FROM foursquare_ext
WHERE latitude IS NOT NULL AND longitude IS NOT NULL
  AND country IS NOT NULL group by country order by cnt_points desc;
```

Type to search

COUNTRY	CNT_POINTS
US	23646375
ID	8568594
TR	8091667
BR	5238774
DE	4984704
JP	4846647
GB	4164084
RU	3088613
FR	3081633
MX	2871667
IT	2838773

Page 1 of 23 (1-11 of 253 items) 1 2 3 4 5 ... 23

Load More

(Without partitioning) Create a table based on external table and add a column with sdo_geometry based on the long and lat columns

```
CREATE TABLE foursquare_geom
NOLOGGING
AS
  SELECT f.*,
  sdo_geometry(2001,4326,sdo_point_type(f.longitude,f.latitude,null),null,null) geometry
  FROM foursquare_ext f
WHERE latitude IS NOT NULL AND longitude IS NOT NULL;
```

NB! Use partitioning: Create a partitioned table based on external table and add a column with sdo_geometry based on the long and lat columns

```
CREATE TABLE foursquare_geom
NOLOGGING
PARALLEL 8
PARTITION BY LIST (country) (
   PARTITION p_us VALUES ('US'),
   PARTITION p_id VALUES ('ID'),
   PARTITION p_tr VALUES ('TR'),
   PARTITION p_br VALUES ('BR'),
   PARTITION p_de VALUES ('DE'),
   PARTITION p_jp VALUES ('JP'),
   PARTITION p_gb VALUES ('GB'),
   PARTITION p_ru VALUES ('RU'),
   PARTITION p_fr VALUES ('FR'),
   PARTITION p_mx VALUES ('MX'),
   PARTITION p_it VALUES ('IT'),
   PARTITION p_th VALUES ('TH'),
   PARTITION p_ca VALUES ('CA'),
   PARTITION p_my VALUES ('MY'),
   PARTITION p_es VALUES ('ES'),
   PARTITION p_kr VALUES ('KR'),
   PARTITION p_pl VALUES ('PL'),
   PARTITION p_be VALUES ('BE'),
   PARTITION p_in VALUES ('IN'),
   PARTITION p_au VALUES ('AU'),
   PARTITION p_nl VALUES ('NL'),
   PARTITION p_se VALUES ('SE'),
   PARTITION p_no VALUES ('NO'),
   PARTITION p_dk VALUES ('DK'),
   PARTITION p_fi VALUES ('FI'),
   PARTITION p_ir VALUES ('IR'),
   PARTITION p_cn VALUES ('CN'),
   PARTITION p_ph VALUES ('PH'),
   PARTITION p_at VALUES ('AT'),
   PARTITION p_others VALUES (DEFAULT)
)
AS
SELECT f.*,
   SDO_GEOMETRY(2001,4326,SDO_POINT_TYPE(f.longitude, f.latitude, NULL), NULL,
NULL) AS geometry
FROM foursquare_ext f
WHERE latitude IS NOT NULL AND longitude IS NOT NULL
AND country IS NOT NULL;
```

Add a primary key to the table

```
ALTER TABLE foursquare_geom
ADD CONSTRAINT pk_foursquare_geom PRIMARY KEY (fsq_place_id);
```

Describe table

```
%script
DESCRIBE foursquare_geom;
```

```
Name                 Null?       Type
FSQ_PLACE_ID         NOT NULL    VARCHAR2(32767)
NAME                             VARCHAR2(32767)
LATITUDE                         NUMBER
LONGITUDE                        NUMBER
ADDRESS                          VARCHAR2(32767)
LOCALITY                         VARCHAR2(32767)
REGION                           VARCHAR2(32767)
POSTCODE                         VARCHAR2(32767)
COUNTRY                          VARCHAR2(32767)
FSQ_CATEGORY_IDS                 JSON
GEOMETRY                         MDSYS.SDO_GEOMETRY

--------------------------
```

Create Spatial index on sdo_geometry column

```
CREATE INDEX fs_geom_idx
ON foursquare_geom(geometry)
INDEXTYPE IS MDSYS.SPATIAL_INDEX_V2
LOCAL;
```

Check if the metadata is created correctly

```
%sql
SELECT
  m.table_name,
  m.column_name,
  d.sdo_dimname,
  d.sdo_lb,
  d.sdo_ub,
  d.sdo_tolerance
FROM
  user_sdo_geom_metadata m,
  TABLE(m.diminfo) d
  where m.table_name='FOURSQUARE_GEOM';
```

Type to search

TABLE_NAME	COLUMN_NAME	SDO_DIMNAME	SDO_LB
FOURSQUARE_GEOM	GEOMETRY	X	-180
FOURSQUARE_GEOM	GEOMETRY	Y	-90

Page 1 of 1 (1-2 of 2 items) 1 Load More

Syntax to create metadata, but MDSYS.SPATIAL_INDEX_V2 creates it for you

```
INSERT INTO user_sdo_geom_metadata
(TABLE_NAME,COLUMN_NAME,DIMINFO,SRID)
VALUES (
 'FOURSQUARE_GEOM','GEOMETRY',
 MDSYS.SDO_DIM_ARRAY(
   MDSYS.SDO_DIM_ELEMENT('longitude', -180, 180, 0.000001),
   MDSYS.SDO_DIM_ELEMENT('latitude', -90, 90, 0.000001)),
 4326 );
```