Contents

[1. CREATE Storage Objects 2](#_Toc498809243)

[2. Generate Test Data in Storage Layers 3](#_Toc498809244)

[2.1.1. Data analysis and preparation 3](#_Toc498809245)

[2.1.2. Source 4](#_Toc498809246)

[2.1.3. Cleansing area 5](#_Toc498809247)

[2.1.4. 3NF 7](#_Toc498809248)

[2.1.5. Result 9](#_Toc498809249)

[3. SQL\*Plus 10](#_Toc498809250)

[3.1.1. Connection 10](#_Toc498809251)

[3.1.2. First select 11](#_Toc498809252)

[3.1.3. Set timing on 11](#_Toc498809253)

[3.1.4. Explain plan for 12](#_Toc498809254)

[3.1.5. Save results 13](#_Toc498809255)

# CREATE Storage Objects

Format Physical Objects of your Business Model according to the templates.

Task Results:

Create scripts for objects:

* Scripts put on Git
* Storage Layers Objects

All scripts were modified accordingly to template and guidance and now stored at dwso->bl\_dm->tables.

# Generate Test Data in Storage Layers

Proceed with loading of Locations data into the normalized level.

Task Results:

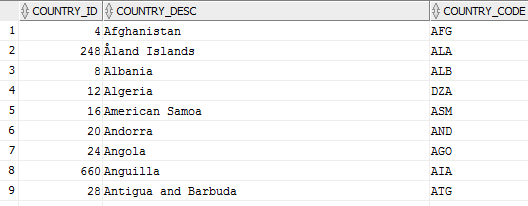
* Scripts put on Git
* Storage Layers Objects test data select screenshots

**Algorithm**

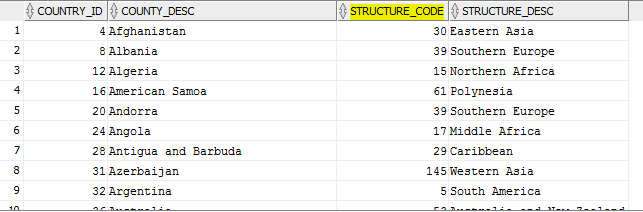
### Data analysis and preparation

1. About data: We have three tables in the denormalized form:

Here just all data about countries.



Here we see table which connects countries with regions. It’s in the 2 NF.

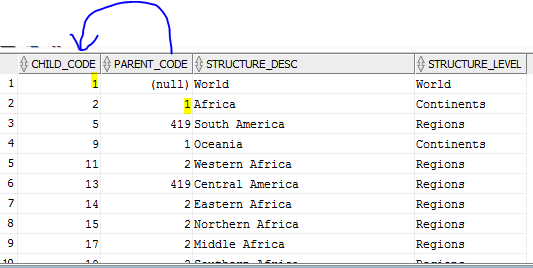


Here we see table which contains and connects 3 levels: World->Continents->Regions.

Parent code in child refers to child code of parent.

(Parent\_Code\_Continent=Child\_Code\_World

Parent\_Code\_Region=Child\_Code\_Continent)



**So, we have the following hierachy:**

World->Continents->Regions->Countries.

1. Created directory on VM (shared folder ETL)

A directory on the server file system is where external binary file LOBs and external table data are located.

CREATE OR REPLACE DIRECTORY etl AS '/media/sf\_ETL';

### Source

1. Created user SRC and gave grants on select, create any table, and read, write from directory

CREATE USER SRC

  IDENTIFIED BY 123456

  DEFAULT TABLESPACE users

  QUOTA UNLIMITED ON users

  TEMPORARY TABLESPACE temp

  PROFILE default;

grant create session to SRC;

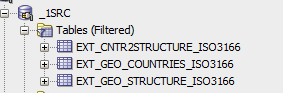
EXEC system.grants\_mgmt.grant\_blat('CREATE ANY TABLE', 'SRC');

EXEC system.grants\_mgmt.grant\_blat('SELECT ANY TABLE', 'SRC');

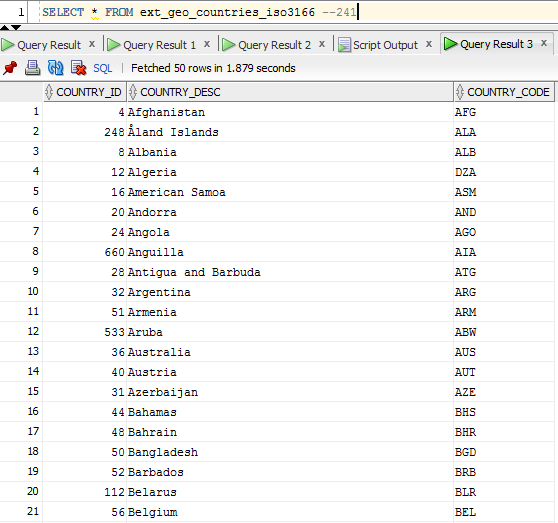
EXEC system.grants\_mgmt.grant\_blat('READ,WRITE ON DIRECTORY ETL', 'SRC');

1. Runned scripts from the task and created 3 tables:

All files (.tab) were put into the directory ETL.



Tables are already loaded with data because external tables enable to access data in external sources as if it were in a table in the database.



### Cleansing area

Here I don’t create wrk\_ and cl\_ tables, because we don’t need to clean or modify our data, therefore it will be enough one table to improve performance (reading from table **is faster** than reading from disk).

But if I did, wrk tables would have the same structure as external tables, and cleansing tables(as I did in this step) are more like the 3NF (but can (not necessarily) have some differences).

1. Created user BL\_CL and gave grants on select, create any table.

CREATE USER BL\_CL

  IDENTIFIED BY 123456

  DEFAULT TABLESPACE users

  QUOTA UNLIMITED ON users

  TEMPORARY TABLESPACE temp

  PROFILE default;

grant create session to BL\_CL;

EXEC system.grants\_mgmt.grant\_blat('CREATE ANY TABLE', 'BL\_CL');

EXEC system.grants\_mgmt.grant\_blat('SELECT ANY TABLE', 'BL\_CL');

1. Created tables, which are close to the 3NF,but without any constraints.

CREATE TABLE cl\_global

(

global\_id NUMBER(10,0),

global\_desc VARCHAR2 ( 200 CHAR )

);

CREATE TABLE cl\_continents

(

continent\_id NUMBER(10,0),

continent\_desc VARCHAR2 ( 200 CHAR ),

global\_id NUMBER(10,0)

);

CREATE TABLE cl\_regions

(

region\_id NUMBER(10,0),

region\_desc VARCHAR2 ( 200 CHAR ),

continent\_id NUMBER(10,0)

);

CREATE TABLE cl\_countries

(

country\_id NUMBER(10,0),

country\_desc VARCHAR2 ( 200 CHAR ),

country\_code VARCHAR2 ( 3 ),

region\_id NUMBER(10,0)

);

1. Gave needed grants, so user BL\_CL can select from SRC tables (run it from the system).

exec system.grants\_mgmt.grant\_blat ('SELECT', 'SRC','ext\_geo\_countries\_iso3166','BL\_CL');

exec system.grants\_mgmt.grant\_blat ('SELECT', 'SRC','ext\_geo\_structure\_iso3166','BL\_CL');

exec system.grants\_mgmt.grant\_blat ('SELECT', 'SRC','ext\_cntr2structure\_iso3166','BL\_CL');

1. Inserted data into them from external tables.

INSERT INTO cl\_global

(global\_id, global\_desc

)

SELECT child\_code,

structure\_desc

FROM src.ext\_geo\_structure\_iso3166

WHERE structure\_level='World';

INSERT INTO cl\_continents

(continent\_id, continent\_desc,global\_id

)

SELECT child\_code,

structure\_desc,

parent\_code

FROM src.ext\_geo\_structure\_iso3166

WHERE structure\_level='Continents';

INSERT INTO cl\_regions

(region\_id, region\_desc,continent\_id

)

SELECT child\_code,

structure\_desc,

parent\_code

FROM src.ext\_geo\_structure\_iso3166

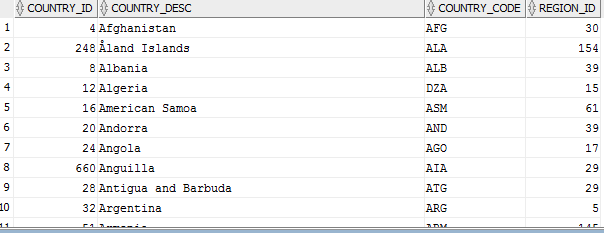
WHERE structure\_level='Regions';

INSERT INTO cl\_countries

(country\_id, country\_code,country\_desc, region\_id

)

COMMIT;



### 3NF

1. Created user BL\_3NF and gave grants on select, create any table.

CREATE USER BL\_3NF

IDENTIFIED BY 123456

DEFAULT TABLESPACE users

QUOTA UNLIMITED ON users

TEMPORARY TABLESPACE temp

PROFILE default;

grant create session to BL\_3NF;

EXEC system.grants\_mgmt.grant\_blat('CREATE ANY TABLE', 'BL\_3NF');

EXEC system.grants\_mgmt.grant\_blat('SELECT ANY TABLE', 'BL\_3NF');

1. Created tables in the 3NF with all required constraints (PK,FK).

* Country code is NULL, because one row (Channel Islands) doesn’t have it.

CREATE TABLE globals

(

global\_id NUMBER(10,0) PRIMARY KEY ,

global\_desc VARCHAR2 ( 200 CHAR ) NOT NULL

);

CREATE TABLE continents

(

continent\_id NUMBER(10,0) PRIMARY KEY ,

continent\_desc VARCHAR2 ( 200 CHAR ) NOT NULL,

global\_id NUMBER(10,0) NOT NULL,

CONSTRAINT gk\_glob FOREIGN KEY (global\_id) REFERENCES globals(global\_id)

);

CREATE TABLE regions

(

region\_id NUMBER(10,0) PRIMARY KEY,

region\_desc VARCHAR2 ( 200 CHAR ) NOT NULL,

continent\_id NUMBER(10,0) NOT NULL,

CONSTRAINT fk\_cont FOREIGN KEY (continent\_id) REFERENCES continents(continent\_id)

);

CREATE TABLE countries

(

country\_id NUMBER(10,0) PRIMARY KEY NOT NULL,

country\_desc VARCHAR2 ( 200 CHAR ) NOT NULL,

country\_code VARCHAR2 ( 3 ),

region\_id NUMBER(10,0)NOT NULL ,

CONSTRAINT fk\_reg FOREIGN KEY (region\_id) REFERENCES regions(region\_id)

);

1. Gave needed grants, so user BL\_3NF can select from BL\_CL tables (run it from system).

exec system.grants\_mgmt.grant\_blat ('SELECT', 'BL\_CL','cl\_continents','BL\_3NF');

exec system.grants\_mgmt.grant\_blat ('SELECT', 'BL\_CL','cl\_countries','BL\_3NF');

exec system.grants\_mgmt.grant\_blat ('SELECT', 'BL\_CL','cl\_global','BL\_3NF');

exec system.grants\_mgmt.grant\_blat ('SELECT', 'BL\_CL','cl\_regions','BL\_3NF');

1. Inserted data into them from BL\_CL.

INSERT INTO globals

SELECT \* FROM bl\_cl.cl\_global;

INSERT INTO continents

SELECT \* FROM bl\_cl.cl\_continents;

INSERT INTO regions

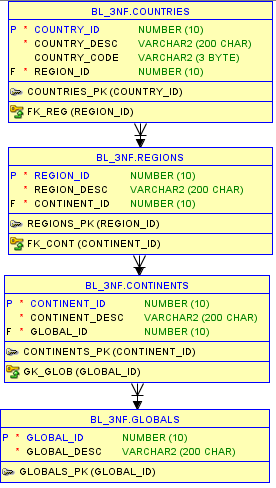
SELECT \* FROM bl\_cl.cl\_regions;

INSERT INTO countries

SELECT \* FROM bl\_cl.cl\_countries;

COMMIT;

### Result



All scripts are also in the folder.

# SQL\*Plus

Connect to a database via SQL\*Plus client and do next steps:

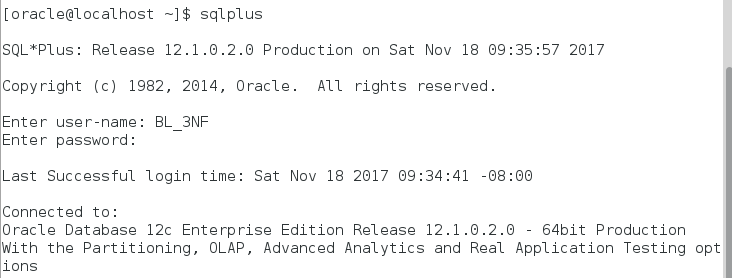
* Show execution plans in SQL\*Plans
* Set timing on
* Run script
* Save data to file

Task Results:

* Put screenshots on Git.

### Connection

Run sqlplus from terminal in Linux. Login and connected as BL\_3NF user.

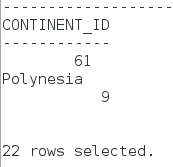
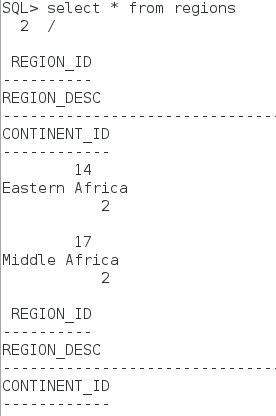




### First select

Run simple select without any additional settings.

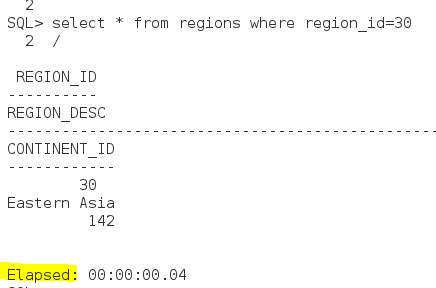
* Sign **‘/’** is used because sqlplus gives an opportunity to write in several lines, and using ‘/’ we show that the command is ready.



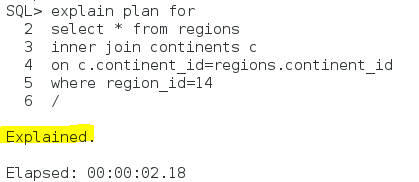
### Set timing on

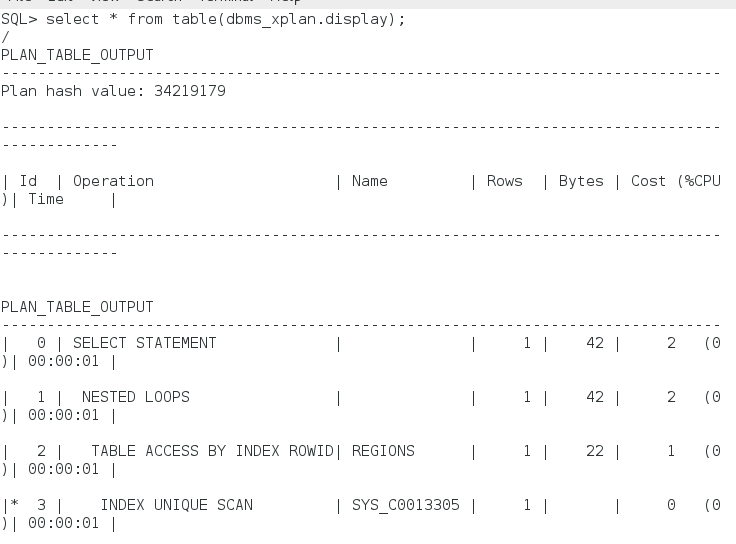
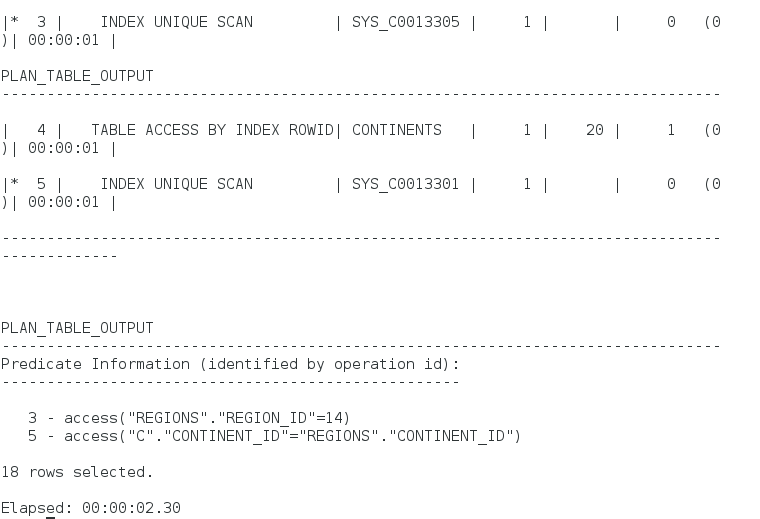


With this option after each query we can see elapsed time (how much time was taken)



### Explain plan for

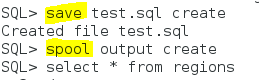


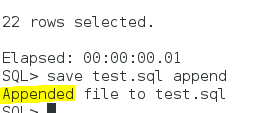
### Save results

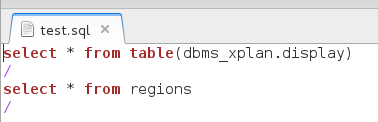
#### Save the query

The last query was saved in the file test.sql



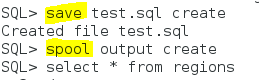
The next query was added to the previous in the file test.sql





#### Save result of the query

The result of the query was saved in the file output.lst



The result of the query was saved in the file spool.txt



All files also in the folder.