|  |
| --- |
| EPAM Systems, RD Dep. |
| Advanced SQL. Core PL SQL |

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| REVISION HISTORY | | | | | |
| Ver. | Description of Change | Author | Date | Approved | |
| Name | Effective Date |
| 1.0 | Initial status | Arina Marchenko | 4-DEC-2017 |  |  |

Contents

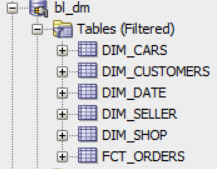
[1. Tables 3](#_Toc500195308)

[2. Bulk collect 3](#_Toc500195309)

[3. FORALL 4](#_Toc500195310)

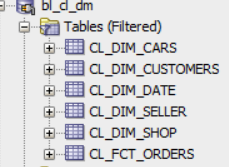
# Tables

On BL\_DM layer there are such tables:



For each dim table we created sequence for the primary key, which were used for linked tables. With it we gave grants to cl\_dm layer for select sequence and select, update, insert and delete on table.

On BL\_CL\_DM layer there are such tables:



Also were used packages.

# Bulk collect

CREATE OR REPLACE PACKAGE pkg\_etl\_load\_ce\_brand

AS

PROCEDURE load\_ce\_brand;

END pkg\_etl\_load\_ce\_brand;

/

CREATE OR REPLACE PACKAGE BODY pkg\_etl\_load\_ce\_brand

AS

PROCEDURE load\_ce\_brand

IS

CURSOR c\_data

IS

SELECT DISTINCT cebr.rowid AS rid,

clbr.brand\_name,

trunc(sysdate) update\_dt

FROM cl\_brand clbr

LEFT OUTER JOIN bl\_3nf.ce\_brand cebr

ON clbr.brand\_name <> cebr.brand\_name;

type t\_\_datafa

IS

TABLE OF c\_data%rowtype INDEX BY binary\_integer;

t\_data t\_\_data;

BEGIN

EXECUTE IMMEDIATE 'truncate table bl\_3nf.ce\_brand';

OPEN c\_data;

LOOP

FETCH c\_data bulk collect INTO t\_data ;

EXIT

WHEN t\_data.count = 0;

FOR idx IN t\_data.first .. t\_data.last

LOOP

IF t\_data(idx).rid IS NULL THEN

INSERT

INTO bl\_3nf.ce\_brand

(

brand\_id,

brand\_name,

update\_dt

)

VALUES

(

bl\_3nf.seq\_brand.nextval,

t\_data(idx).brand\_name,

sysdate

);

ELSE

UPDATE bl\_3nf.ce\_brand

SET update\_dt = sysdate

WHERE rowid = t\_data(idx).rid ;

END IF;

END LOOP;

END LOOP;

CLOSE c\_data;

COMMIT;

EXCEPTION

WHEN OTHERS THEN

RAISE;

END load\_ce\_brand;

END pkg\_etl\_load\_ce\_brand;

/

# FORALL

CREATE OR REPLACE PACKAGE BODY pkg\_etl\_load\_cl\_tables

AS

PROCEDURE load\_cl\_dim\_cars

IS

BEGIN

DECLARE

CURSOR c\_data IS

SELECT c.car\_id,

c.car\_number,

c.car\_name,

vt.vehicle\_type\_name,

et.engine\_type\_name,

gt.gearbox\_type\_name,

m.model\_name,

rs.repaired\_status\_name,

c.start\_dt,

c.end\_dt

FROM bl\_3nf.ce\_cars c

left join bl\_3nf.ce\_vehicle\_type vt on c.vehicle\_type\_id = vt.vehicle\_type\_id

left join bl\_3nf.ce\_engine\_type et on c.engine\_type\_id=et.engine\_type\_id

left join bl\_3nf.ce\_gearbox\_type gt on c.gearbox\_type\_id = gt.gearbox\_type\_id

left join bl\_3nf.ce\_model m on c.model\_id=m.model\_id

left join bl\_3nf.ce\_repaired\_status rs on c.repaired\_status\_id = rs.repaired\_status\_id;

TYPE fetch\_array IS TABLE OF c\_data%ROWTYPE ;

s\_array fetch\_array;

BEGIN

OPEN c\_data;

LOOP

FETCH c\_data BULK COLLECT INTO s\_array;

FORALL i IN 1..s\_array.COUNT

INSERT INTO fct\_sales

VALUES s\_array(i) ;

EXIT WHEN c\_data%NOTFOUND;

END LOOP;

CLOSE c\_data;

COMMIT;

END load\_cl\_dim\_cars;