1. Automation of diff and load procedures **referring to package** “OWNER\_CORE.LIB\_GENERATOR” – **parameters are filled by user** so you do not need to write these procedures on your own. It is highly recommended to use this code so **every package is standardized –** checkDWH solution in package “OWNER\_ETL.LIB\_DCT\_APPLICANT”:

Text, letter

Description automatically generated

**NOTE: schema for this generator is different = OWNER\_ETL, but for DM / LDM purposes you use OWNER\_CORE!**

1. General recommendations

* Each **code part representing some business logic** should be **shortly described** by comment inside code for future adjustments so other developers understand what is going on there
* Write executable queries to merge requests = **always check that written query works**
* Don’t use reference on AP or OWNER\_INT or OWNER\_STG tables/views/other objects = replace these objects for either OWNER\_DWH or DM/LDM objects
* **Use OWNER\_DWH tables instead of views** so you will avoid possible data masking issues
* **Follow** [DWH naming convention](https://wiki.homecredit.net/confluence/pages/viewpage.action?spaceKey=DWH&title=Naming+Conventions%2C+Attribute+Standards)
* try to avoid using SKP / SKF columns as much as possible:
  + Use business columns when merging dimensional tables if possible
  + **Don’t use SKP** key columns in WHERE clause = find attribute or group of attributes that should be filtered and use them instead
* **Do not use SYSDATE** (except views but be aware of functionality). Instead of that use DATE\_EFFECTIVE
* If you develop **OWNER\_xxx** objects use **HS\_2016.5** Project on Git
* **Use WITH READ ONLY** When creating views. Exception are OWNER\_EXT views.
* **View should be 1:1 with referenced table** so that means that all data transformations should be done within table’s package
* **Each table/view and its attributes** (columns) must have **proper description** which briefly describes the content of the table/view and attribute (column)
* Stage tables **STM and STC** should **not have NOT NULL constraints or indexes**
* **Separate ordering of audit columns and business columns** when creating tables
* **SKP/SKF and CNT** columns should be always with data format **INTEGER** due to fact that **INTEGER is constrained** while NUMBER is not so INTEGER takes **less space**