

Exercise #1: Dynamic query parameterized with another query result

- Write a stored procedure that counts the number of rows in all tables in your system catalog (system view `TAB (TNAME, TABTYPE...)`).
- Exclude views (type `VIEW`) from the result.
- Display the result ordered by alphabetical order under the form `TABLE_NAME: NB_TUPLES tuple(s)`.
- Handle the plural of the word “tuple”, which takes an “s” only when table size is strictly greater than 1.
- Test!

Exercise #2: Parameterized schema alteration

1. In an anonymous PL/SQL block, define a string variable named *source* and initialize it with the name of a table present on your account. Define a second string variable named *destination* initialize it with any value (e.g., `COPY`). In the code section, program the copy of table *source* into table *destination* (creation of table *destination* by selecting all attributes and tuples from table *source*). Test!
2. Transform your anonymous PL/SQL block into a stored procedure named *copytable*, with *source* and *destination* becoming parameters. Does it work?
3. Add the `AUTHID CURRENT_USER` clause in the procedure definition, after defining the parameters (e.g., `CREATE OR REPLACE PROCEDURE copytable(source VARCHAR, destination VARCHAR) AUTHID CURRENT_USER IS`). Test! What has changed? Why was this manipulation superfluous when you created stored procedures before?

Exercise #3: Dynamic query parameterized with another query result (again!)

Write a stored procedure that retrieves tables containing an attribute whose name contains a string passed as parameter; as well as the name, type and number of distinct values of this attribute in the table.

Exercise #4: Parameterized view creation

- Write a stored procedure that inputs a table name and creates a view containing the name and type of all attributes in the input table.
- The name of the view must be formatted as `ATT_table_name`.
- Use system view `USER_TAB_COLUMNS (TABLE_NAME, COLUMN_NAME, DATA_TYPE...)` to retrieve attributes' name and type.
- Check-up the result!

Note: In system views, all strings (such as table or attribute names) are stored in large caps.