

# Contents

Chapter 1 What is Miqualat Database .....	1
Chapter 2 Setting Up the Working Environment .....	2
2.1 Install Jupyter Notebook.....	2
2.2 Install MySQL Server.....	2
2.3 Install MySQLclient .....	3
2.4 Install Required Python Libraries .....	3
Chapter 3 Folder and Content Description .....	4
3.1 Description of the FUNCTIONS Folder .....	4
3.2 Description of the INPUT Folder .....	4
3.3 Description of the OUTPUT Folder .....	5
3.4 Description of the DATABASE_SQL Folder .....	5
Chapter 4 Miqualat Notebooks Description and Usage .....	6
Chapter 5 Create Biomart File.....	8
Chapter 6 Create File Table.....	10
Chapter 7 Database Description .....	12
7.1 Database Schema.....	12
7.2 Database Description.....	13
7.3 Checks Description.....	16
Chapter 8 Database Export Data .....	17

## Chapter 1 What is Miqualat Database

---

With Miqualat Notebooks you have access to an easy-to-use Jupyter Notebook interface.

Miqualat Notebook allows you to save data from publications, genes, variants, technique and other, associate them with descriptions and tags, import/export data in the Miqualat Database.

A control system has been implemented to maintain flexibility and freedom to import data while maintaining control over them.

You can get this data from different databases as pubmed, ensembl, kegg and make automatically links between the entries.

This relational database allows you to link and export information from different tables using a specific Jupyter Notebook.

## Chapter 2 Setting Up the Working Environment

### 2.1 Install Jupyter Notebook

---

Update and upgrade your system (recommended).

```
$ sudo apt update && sudo apt upgrade
```

Install Jupyter Notebook.

```
$ sudo apt install jupyter-notebook
```

If you need to configure the conf file this is located in:

```
/root/.jupyter/jupyter_notebook_config.py
```

```
$ sudo jupyter-notebook --generate-config
```

To access Jupyter Notebook you may need to set up a password.

```
$ sudo jupyter-notebook password
```

Run jupyter notebook (on port 8888 by default).

```
$ sudo jupyter-notebook
```

### 2.2 Install MySQL Server

---

Install apache2 and mysql-server.

```
$ sudo apt install apache2
```

```
$ sudo apt install mysql-server
```

TIP: If you use raspberry or a similar system install mariadb and configure it.

```
$ sudo apt install mariadb-server-10.0
```

Create user and set password for database (this user can to access in phpMyAdmin, keep your credentials).

```
$ sudo mysql -u root
```

```
> CREATE USER 'user'@'localhost' IDENTIFIED BY 'password';
```

Grant privileges on all databases.

```
> GRANT ALL PRIVILEGES ON * . * TO 'user'@'localhost';
```

Grant privileges on one databases (safer).

```
> GRANT ALL PRIVILEGES ON DATABASE. * TO 'user'@'localhost';
```

```
> FLUSH PRIVILEGES;
```

!! IMPORTANT (pay attention before continuing with the command below), when you install phpMyAdmin ,select apache2 (with TAB) and press SPACE (for select apache2), after leave all empty (press ENTER) !!

```
$ sudo apt install phpmyadmin
```

```
$ sudo systemctl restart apache2
```

Connect phpmyadmin using your browser (on port 80: /phpmyadmin).

[http://192.168.0.\(continue\\_your\\_ip\).../phpmyadmin](http://192.168.0.(continue_your_ip).../phpmyadmin)

Show your IP.

```
$ hostname -I
```

<http://localhost/phpmyadmin> (work only on local computer not for remote server)

## 2.3 Install MySQLclient

---

Install mysqlclient (connect database remotely using python).

Get more information at: <https://pypi.org/project/mysqlclient/>

```
$ sudo apt-get install python3-dev default-libmysqlclient-dev build-essential
```

```
$ pip3 install mysqlclient
```

Update and upgrade your system (recommended).

```
$ sudo apt update && sudo apt upgrade
```

## 2.4 Install Required Python Libraries

---

Install python3-pip for install python3 libraries.

```
$ sudo apt install python3-pip
```

The following libraries are required for use miqualat notebooks.

```
$ sudo pip3 install biopython
```

```
$ sudo pip3 install ensembl-rest
```

```
$ sudo pip3 install mygene
```

```
$ sudo pip3 install pandas
```

```
$ pip3 install reactome2py
```

## Chapter 3 Folder and Content Description

### 3.1 Description of the FUNCTIONS Folder

---

The functions in FUNCTIONS folder, contain python code that is recalled in the notebooks.

1. `ensembl_search.py`  
Use the `ensembl-rest` library to get information about genes from the Ensembl database, for GENE table.
2. `ensembl_to_kegg_id.py`  
Convert Ensembl gene id to the Kegg id of the gene and related pathways, using `mygene` and `Bio.KEGG.REST` (in `biopython` library), for GEN\_KEGG table.
3. `pubmed_search.py`  
Carry out searches within the pubmed database using `biopython`, for PUBLICATION table.
4. `python_parser_biomart_gene_csv.py`  
Convert the biomart file (which contains all biomart genes) into the right format, for GENE table.
5. `MIQUALAT_manual_table.py`  
Allows you to create non-automated tables (PUB\_GEN\_TEC\_VAR\_TAG, TAG, TECNQUE, VARIANT). It's based on loops who add one line at time at csv (when you enter yes the function repeat the loop or save file to the INPUT folder if you leave empty), there are an option to simplify this process and insert much genes at time to PUB\_GEN\_TEC\_VAR\_TAG table.  
This function has a control on NULL value (convert all empty value in NULL if possible, in any case NULL is converted to uppercase). There are automatically warning if you cannot enter NULL value.
6. `MIQUALAT_data_import_and_check.py`  
Import and check the csv files processed with previous notebooks, using `MySQLdb` (in `mysqlclient` library).
7. `MIQUALAT_data_export.py`  
Allows you to export (to csv) or view database data through mysql queries. The function uses `pandas` for data representation.
8. `reactome_pathfinder.py`  
Get information from Reactome database.

### 3.2 Description of the INPUT Folder

---

In the INPUT folder are saved all the files generated by the notebooks, ready to be imported into the miqualat database.

Generated by:

- MIQUALAT\_GENE\_data\_assess.ipynb
- MIQUALAT\_DB\_for\_KEGG\_data\_assess.ipynb
- MIQUALAT\_GEN\_DB\_for\_KEGG.ipynb
- MIQUALAT\_PUBLICATION\_data\_assess.ipynb
- MIQUALAT\_manual\_table.ipynb
- MIQUALAT\_gene\_ontology\_notebook.ipynb
- MIQUALAT\_reactome\_notebook.ipynb

Imported by:

- MIQUALAT\_data\_import\_and\_check.ipynb

### 3.3 Description of the OUTPUT Folder

---

In the OUTPUT folder are saved all files exported by the notebook from miqualat database.

Exported by:

- MIQUALAT\_data\_export.ipynb

### 3.4 Description of the DATABASE\_SQL Folder

---

The DATABASE\_SQL folder contains a copy of database:

MIQUALAT\_create\_database\_and\_tables.sql.

## Chapter 4 Miqualat Notebooks Description and Usage

---

Notebooks are designed to simplify the generation, import, export of information from the database. To use notebooks select the code cell and click on run, below will describe the notebooks and their functions.

Five notebooks have been built, that allow you to search, import and check into the miqualat database.

1. MIQUALAT\_PUBLICATION\_data\_assess.ipynb  
Carry out searches within the pubmed database, automatize data collection for PUBLICATION table.  
Creates PUBLICATION.csv file and move all searches (PUBMED\_SEARCH\_DATA\_Y-m-d\_H-M-S.txt format) in the OUTPUT folder.
2. MIQUALAT\_DB\_for\_KEGG\_data\_assess.ipynb  
Download genes and pathways from kegg database (from kegg code), automatize data collection for KEGG table.  
Creates ALL\_KEGG\_GENE.csv (all kegg genes from kegg code) in the OUTPUT folder.  
Creates ALL\_KEGG\_PATH.csv (all kegg pathways from kegg code) in the OUTPUT folder.
3. MIQUALAT\_GENE\_data\_assess.ipynb  
Allows you to download genes from ensembl database, automatize data collection for GENE table.  
Option 1) get information from specific genes (ensembl gene id required).  
Option 2) convert all genes downloaded from biomart (GENE.csv required).  
In any case creates GENE.csv and move it in the OUTPUT folder.
4. MIQUALAT\_GEN\_DB\_for\_KEGG.ipynb  
Correlates the ensembl gene id to the kegg id (if possible) of the gene and related pathway, automatize data collection for GEN\_KEGG table.  
Option 1) correlates the ensembl gene id to the kegg id of the gene and related pathways for specific gene (ensembl gene id required), for specific gene.  
Option 2) processes the entire GENE.csv file.  
Creates ENS\_GENE\_ID.csv (Ensembl gene id to Kegg gene id) in the OUTPUT folder.  
Creates ENS\_PATH\_ID.csv (Ensembl gene id to Kegg pathways id) in the OUTPUT folder.
5. MIQUALAT\_manual\_table.ipynb  
This notebook use the MIQUALAT\_manual\_table.py function in FUNCTIONS folder.  
Allows you to create non-automated tables (PUB\_GEN\_TEC\_VAR\_TAG, TAG, TECNQUE, VARIANT). It's based on loops who add one line at time at csv (when you enter yes the function repeat the loop or save file to the INPUT folder if you leave empty),

there are an option to simplify this process and insert much genes at time to PUB\_GEN\_TEC\_VAR\_TAG table.

This function have a control on NULL value (convert all empty value in NULL if possible, in any case NULL is converted to uppercase). There are automatically warning if you cannot enter NULL value.

Follow the instructions to create the desired table.

6. MIQUALAT\_data\_import\_and\_check.ipynb

Allows the import of files processed with previous notebooks, once checked.

An error control system prevents the import of incorrect information.

7. MIQUALAT\_data\_export.ipynb

Allows the export data from Miqualat database.

8. MIQUALAT\_gene\_ontology\_notebook.ipynb

Correlates the ensembl gene id or list of ensembl gene ids to all the gene ontology terms related to the ensembl ids in the entries list;

and creates the input files GEN\_DB.csv and DB.csv to import gene ontology repository data in miqualat database.

9. MIQUALAT\_reactome\_notebook.ipynb

Correlates the ensembl gene id or list of ensembl gene ids to all reactome pathway codes related to the ensembl ids in the entries list;

and creates the input files GEN\_DB.csv and DB.csv to import gene ontology repository data in miqualat database.



## Chapter 5 Create Biomart File

Go to <https://www.ensembl.org/biomart/martview/afe982e758c87b06e672bc93a42a4f30>.

Export all results to csv file.

Select dataset Ensembl Gene 101.

Select Species (refseq) (ex Cow genes (ARS-UCD1.2)).

Go to Attributes, Gene and create the biomart file following this header (select in the right order).

**Gene stable ID, Gene name, Gene description, Chromosome/scaffold name, Gene start (bp), Gene end (bp), Strand**

The screenshot shows the Ensembl BioMart interface. At the top, the Ensembl logo is followed by links for BLAST/BLAT, VEP, Tools, and BioMart. Below this is a navigation bar with 'New', 'Count', and 'Results' tabs. The main interface is divided into two columns. The left column contains a 'Dataset' dropdown set to 'Ensembl Genes 101', a 'Filters' section with '[None selected]', and an 'Attributes' section with a list of attributes: 'Gene stable ID', 'Gene stable ID version', 'Transcript stable ID', and 'Transcript stable ID version'. Below this is another 'Dataset' dropdown set to '[None Selected]'. The right column contains a 'Structures' section with a radio button selected for 'Sequences' and a radio button for 'Homologues (Max select 6 orthologues)'. Below this is a 'GENE:' section with a list of attributes under the 'Ensembl' heading. The attributes are: 'Gene stable ID' (checked), 'Gene stable ID version' (unchecked), 'Transcript stable ID' (unchecked), 'Transcript stable ID version' (unchecked), 'Protein stable ID' (unchecked), 'Protein stable ID version' (unchecked), 'Exon stable ID' (unchecked), 'Gene description' (checked), 'Chromosome/scaffold name' (checked), 'Gene start (bp)' (checked), 'Gene end (bp)' (checked), 'Strand' (checked), and 'Karyotype band' (unchecked). To the right of these are several other attributes: 'Transcript length (including)' (unchecked), 'APPRIS annotation' (unchecked), 'Gene name' (checked), 'Source of gene name' (unchecked), 'Transcript name' (unchecked), 'Source of transcript name' (unchecked), 'Transcript count' (unchecked), 'Gene % GC content' (unchecked), 'Gene type' (unchecked), 'Transcript type' (unchecked), 'Source (gene)' (unchecked), 'Source (transcript)' (unchecked), and 'Version (gene)' (unchecked).

**Ensembl**

- ☒ Gene stable ID
- ☐ Gene stable ID version
- ☐ Transcript stable ID
- ☐ Transcript stable ID version
- ☐ Protein stable ID
- ☐ Protein stable ID version
- ☐ Exon stable ID
- ☒ Gene description
- ☒ Chromosome/scaffold name
- ☒ Gene start (bp)
- ☒ Gene end (bp)
- ☒ Strand
- ☐ Karyotype band

☐ Transcript length (including)

☐ APPRIS annotation

☒ Gene name

☐ Source of gene name

☐ Transcript name

☐ Source of transcript name

☐ Transcript count

☐ Gene % GC content

☐ Gene type

☐ Transcript type

☐ Source (gene)

☐ Source (transcript)

☐ Version (gene)

Results

[★ URL](#)
[XML](#)
[Perl](#)
[Help](#)

CD1.2)

Export all results to

File

CSV

☐ Unique results only

Go

Email notification to

View

10

rows as

HTML

☐ Unique results only

Gene stable ID	Gene name	Gene description	Chromosome/scaffold name	Gene start (bp)	Gene end (bp)	Strand
<a href="#">ENSBTAG00000006648</a>			<a href="#">1</a>	<a href="#">339070</a>	<a href="#">350389</a>	-1
<a href="#">ENSBTAG00000049697</a>	<a href="#">5S_rRNA</a>	5S ribosomal RNA [Source:RFAM;Acc:RF00001]	<a href="#">1</a>	<a href="#">475398</a>	<a href="#">475516</a>	1

## Chapter 6 Create File Table

---

To pass the check, csv must have the following headers.

**!! CREATE THE TABLE FILE FOLLOWING THIS HEADERS (CREATE THE HEADER AS IT IS WRITTEN, DO NOT ADD QUOTES OR ANYTHING ELSE) !!**

1. PUB\_GEN\_TEC\_VAR\_TAG TABLE  
**integer\_progressive\_ID, pubmed\_ID, ensembl\_gene\_ID, variant\_name, technique, keyword\_tags, relationship\_note**
2. PUBLICATION TABLE  
**pubmed\_ID, doi, article\_title, article\_authors, article\_journal, publication\_year**
3. TECHNIQUE TABLE  
**technique, technique\_short\_description**
4. TAG TABLE  
**keyword\_tags, tags\_short\_description**
5. GENE TABLE  
**ensembl\_gene\_ID, gene\_name, gene\_short\_description, refseq, species, chromosome, start\_coordinate, end\_coordinate, strand**
6. DB TABLE  
**database\_ID, database\_object\_type, database\_object\_name**
7. GEN\_DB TABLE  
**ensembl\_gene\_ID, database\_ID**
8. VARIANT TABLE  
**variant\_name, variant\_type, chromosome, position, reference\_allele, alternative\_allele, rs\_ID, species, refseq**

We prosed as variant\_name this combination of values

- 1) international code for species/reference sequence (i.e. hg38 for last human reference sequence)
- 2) chromosome
- 3) position

4) reference allele

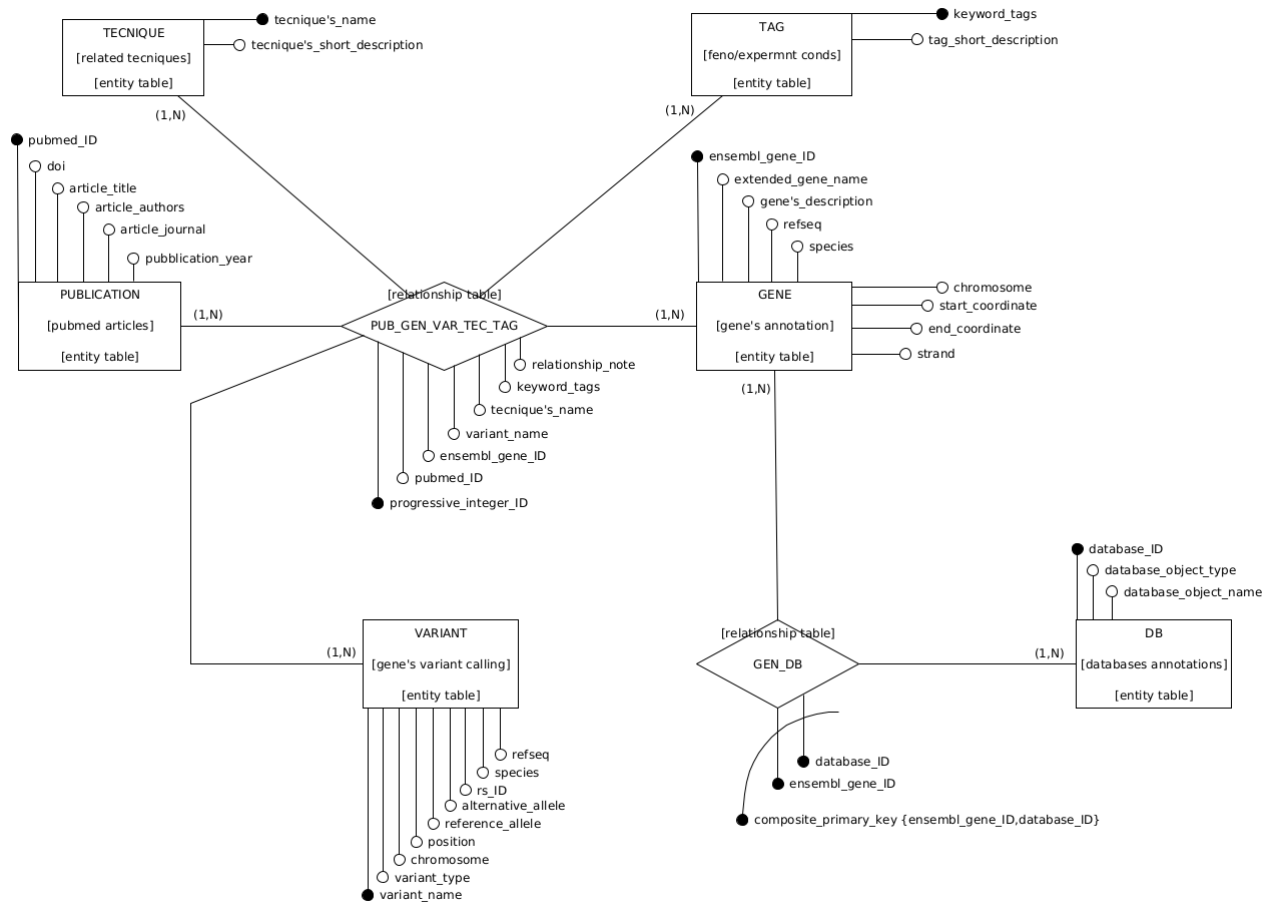
5) alternative allele (i.e. if the alternative allele is too long, try to find a way to summarize it)

example: hg38\_1:146793\_A|G

## Chapter 7 Database Description

### 7.1 Database Schema

Below the physical Entity-Relationship diagram (ERD) of the Miqualat database, this describe the relations between the database tables.



## 7.2 Database Description

---

Below is an accurate description of all the tables in the database.

```
mysql> SHOW TABLES;
```

Tables_in_MIQUALAT
DB
GENE
GEN_DB
PUBLICATION
PUB_GEN_VAR_TEC_TAG
TAG
TECHNIQUE
VARIANT

8 rows in set (0.00 sec)

```
##description of the structure of the tables and the type of data  
of the attributes of each entity and relation##
```

```
mysql> DESCRIBE GENE;
```

Field	Type	Null	Key	Default	Extra
ensembl_gene_ID	varchar(20)	NO	PRI	NULL	
gene_name	varchar(20)	YES		NULL	
gene_short_description	varchar(300)	YES		NULL	
refseq	varchar(50)	NO		NULL	
species	varchar(50)	NO		NULL	
chromosome	tinyint	NO		NULL	
start_coordinate	int unsigned	NO		NULL	
end_coordinate	int unsigned	NO		NULL	
strand	tinyint	NO		NULL	

9 rows in set (0.00 sec)

```
mysql> DESCRIBE GEN_DB;
```

Field	Type	Null	Key	Default	Extra
ensembl_gene_ID	varchar(20)	NO	PRI	NULL	
database_ID	varchar(20)	NO	PRI	NULL	

2 rows in set (0.01 sec)

```
mysql> DESCRIBE DB;
```

Field	Type	Null	Key	Default	Extra
database_ID	varchar(50)	NO	PRI	NULL	
database_object_type	varchar(200)	NO		NULL	
database_object_name	varchar(500)	YES		NULL	

3 rows in set (0.00 sec)

```
mysql> DESCRIBE PUBLICATION;
```

Field	Type	Null	Key	Default	Extra
pubmed_ID	int unsigned	NO	PRI	NULL	
doi	varchar(100)	YES		NULL	
article_title	varchar(300)	NO		NULL	
article_authors	varchar(300)	NO		NULL	
article_journal	varchar(100)	NO		NULL	
publication_year	year	NO		NULL	

```
6 rows in set (0.00 sec)
```

```
mysql> DESCRIBE PUB_GEN_VAR_TEC_TAG;
```

Field	Type	Null	Key	Default	Extra
integer_progressive_ID	int unsigned	NO	PRI	NULL	auto_increment
pubmed_ID	int unsigned	NO	MUL	NULL	
ensembl_gene_ID	varchar(20)	YES	MUL	NULL	
variant_name	varchar(30)	YES	MUL	NULL	
technique	varchar(50)	YES	MUL	NULL	
keyword_tags	varchar(50)	YES	MUL	NULL	
relationship_note	varchar(200)	YES		NULL	

```
7 rows in set (0.01 sec)
```

```
mysql> DESCRIBE TAG;
```

Field	Type	Null	Key	Default	Extra
keyword_tags	varchar(50)	NO	PRI	NULL	
tags_short_description	varchar(200)	NO		NULL	

```
2 rows in set (0.01 sec)
```

```
mysql> DESCRIBE TECHNIQUE;
```

Field	Type	Null	Key	Default	Extra
technique	varchar(50)	NO	PRI	NULL	
technique_short_description	varchar(300)	NO		NULL	

```
2 rows in set (0.01 sec)
```

```
mysql> DESCRIBE VARIANT;
```

Field	Type	Null	Key	Default	Extra
variant_name	varchar(50)	NO	PRI	NULL	
variant_type	varchar(50)	NO		NULL	
chromosome	tinyint unsigned	NO		NULL	
position	int unsigned	NO		NULL	
reference_allele	varchar(20)	NO		NULL	
alternative_allele	varchar(20)	NO		NULL	
rs_ID	varchar(20)	YES		NULL	
species	varchar(50)	NO		NULL	
refseq	varchar(50)	NO		NULL	

```
9 rows in set (0.00 sec)
```

```
mysql> exit
```

```
Bye
```

##SI TRATTA DI UN DATABASE RELAZIONALE##

\*CON 6 TABELLE DI ENTITÀ{

```
-----PUBLICATION,
-----GENE,
-----VARIANT,
-----TECHNIQUE,
-----TAG,
-----DB-----
}
```

\*E 2 TABELLE RELAZIONALI{

```
-----PUB_GEN_VAR_TEC_TAG,
-----GEN_DB
}
```



## 7.3 Checks Description

---

Import checks types:

- headers check;
- entry field number check;
- duplicate entry check;
- int data type field check;
- primary keys check;
- foreign keys check;
- NULL value field check;

## Chapter 8 Database Export Data

---

MIQUALAT\_data\_export.ipynb notebook allows you to export data from miqualat database.

To export the file run the cell, enter the desired query number and follow the instructions.

Files are saved in OUTPUT folder as:

table\_TABLE\_query\_number\_NUMBER\_export\_data\_results\_\_Y-m-d\_h-d-s.

tables selection menu from which to export data:

```
enter 1 to export data from database MIQUALAT following table: GENE;
enter 2 to export data from database MIQUALAT following table: KEGG;
enter 3 to export data from database MIQUALAT following table: GEN_KEGG;
enter 4 to export data from database MIQUALAT following table: PUBLICATION;
enter 5 to export data from database MIQUALAT following table: TECNIQUE;
enter 6 to export data from database MIQUALAT following table: TAG;
enter 7 to export data from database MIQUALAT following table: VARIANT;
enter 8 to export data from database MIQUALAT following table: PUB_GEN_VAR_TEC_TAG;
```

queires selection menu to export data from table GENE:

```
1: to extract all table records;
2: to extract specific record relative to an input ensembl ID;
3: to extract specific record relative to an input gene_name;
4: to extract all the species and relative refseq version;
5: to count genes total numbers sorted by species;
6: to extract all genes from an input species;
7: to extract all genes located on an input chromosome for an input species;
8: to extract all genes located between start and end input coordinates on an input chromosome for an input species;
```

queires selection menu to export data from table KEGG:

```
1: to extract all table records;
2: to extract all record fields relative to an input kegg_ID;
3: to extract all record fields relative to an input compound name (or pathway or molecule) to search into field kegg_object_name;
```

queires selection menu to export data from table GEN\_KEGG:

```
1: to extract all table records;
2: to extract kegg_id,ensembl_gene_ID,gene_name,species of all genes relative to an input compound name (or pathway or molecule) to search into field kegg_object_name;
3: to extract all KEGG record fields related to an input species and an input kegg_object_type (gene, pathway, protein, enzyme or others);
```

queires selection menu to export data from table PUBLICATION:

```
1: to extract all table records;
2: to extract all record fields of a paper relative to an input pubmed_ID;
3: to extract all record fields of a paper relative to an input doi;
4: to extract all record fields of all papers relative to an input journal;
5: to extract all record fields of all papers relative to an input author to search into field author_s_name;
6: to extract all record fields of all papers relative to an input publication year date;
```

queires selection menu to export data from table TECNQUE:

- 1: to extract all table records;
- 2: to extract all record fields relative to an input tecnique;
- 3: to extract all record fields relative to an input term to search into field tecnique\_short\_description;

queires selection menu to export data from table TAG:

- 1: to extract all table records;
- 2: to extract all record fields relative to an input keyword\_tag;
- 3: to extract all record fields relative to an input term to search into field keyword\_tags;
- 4: to extract all record fields relative to an input term to search into field tags\_short\_description;

queires selection menu to export data from table VARIANT:

- 1: to extract all table records;
- 2: to extract all record fields relative to an input variant\_name;
- 3: to extract all record fields of all variants relative to an input ensembl\_gene\_ID;
- 4: to extract all record fields of all variant relative to an input gene\_name of an input species;
- 5: to extract all record fields of all variants and the respective ensembl\_gene\_ID and gene\_name relative to an input species;
- 6: to extract all record fields of all variants and the respective ensembl\_gene\_ID and gene\_name relative to an input species and an input chromosome;

queires selection menu to export data from table PUB\_GEN\_VAR\_TEC\_TAG:

- 1: to extract all table records;
- 2: to extract pubmed\_ID,doi,article\_title of all publications related to an input ensembl\_gene\_ID;
- 3: to extract pubmed\_ID,doi,article\_title of all publications related to an input gene\_name;
- 4: to extract ensembl\_gene\_ID and gene\_name of all genes related to an input pubmed\_ID;
- 5: to extract ensembl\_gene\_ID and gene\_name of all genes related to an input doi;
- 6: to extract pubmed\_ID,doi,article\_title and ensembl\_gene\_ID and gene\_name of all publications and genes related to an input keyword\_tag;
- 7: to extract ensembl\_gene\_ID and gene\_name of all genes related to an input variant\_name;
- 8: to extract all tecnique informations of all tecniques related to an input pubmed\_ID;
- 9: to extract all kegg\_ID informations of all kegg\_IDs (gene, pathway, protein, enzyme...etc.) related to an input pubmed\_ID;
- 10: to extract pubmed\_ID,doi,article\_title,ensembl\_gene\_ID and gene\_name of all publications and genes related to an input kegg\_ID (kegg code of gene, pathway, protein, enzyme...etc.);
- 11: to extract pubmed\_ID,doi,article\_title,ensembl\_gene\_ID and gene\_name of all publications and genes related to an input compound name to search into kegg\_object\_name;
- 12: to extract all record fields and gene\_name related to an input pubmed\_ID;
- 13: to extract all record fields and gene\_name related to an input doi;
- 14: to extract pubmed\_ID,doi,article\_title and tecnique\_name and keyword\_tag related to an input ensembl\_gene\_ID;
- 15: to extract pubmed\_ID,doi,article\_title and tecnique\_name and keyword\_tag related to an input gene\_name;