Contents

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

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
```

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 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.

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.

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_KEGG_data_assess.ipynb
- MIQUALAT_PUBLICATION_data_assess.ipynb
- MIQUALAT_manual_table.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 PUBLICAITON 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_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_KEGG_data_assess.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),

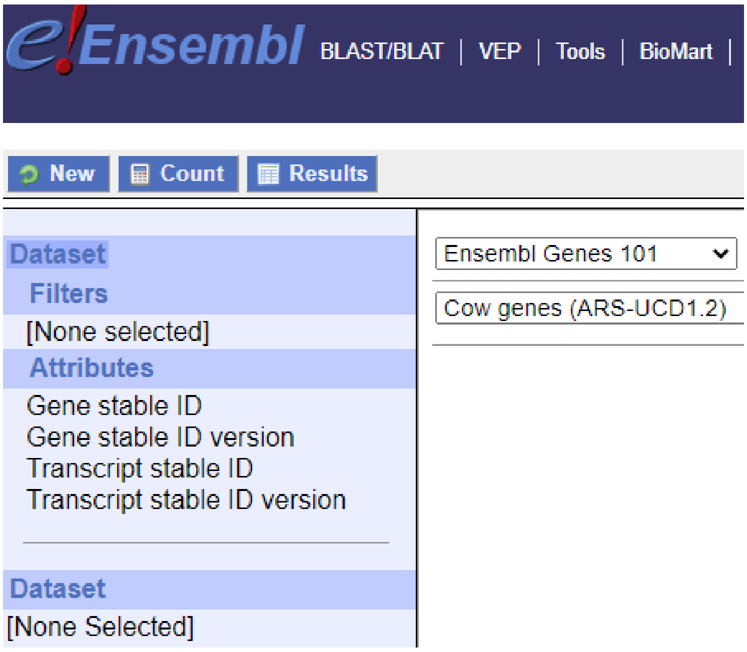
there are an option to simplify this process and insert much genes at time to PUB_GEN_TEC_VAR_TAG table.
This function have a control on NULL value (convert all empty value in NULL if possible, in any case NULL is converted to uppercase). There are automatically warning if you cannot enter NULL value.
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.

Chapter 5 Create Biomart File

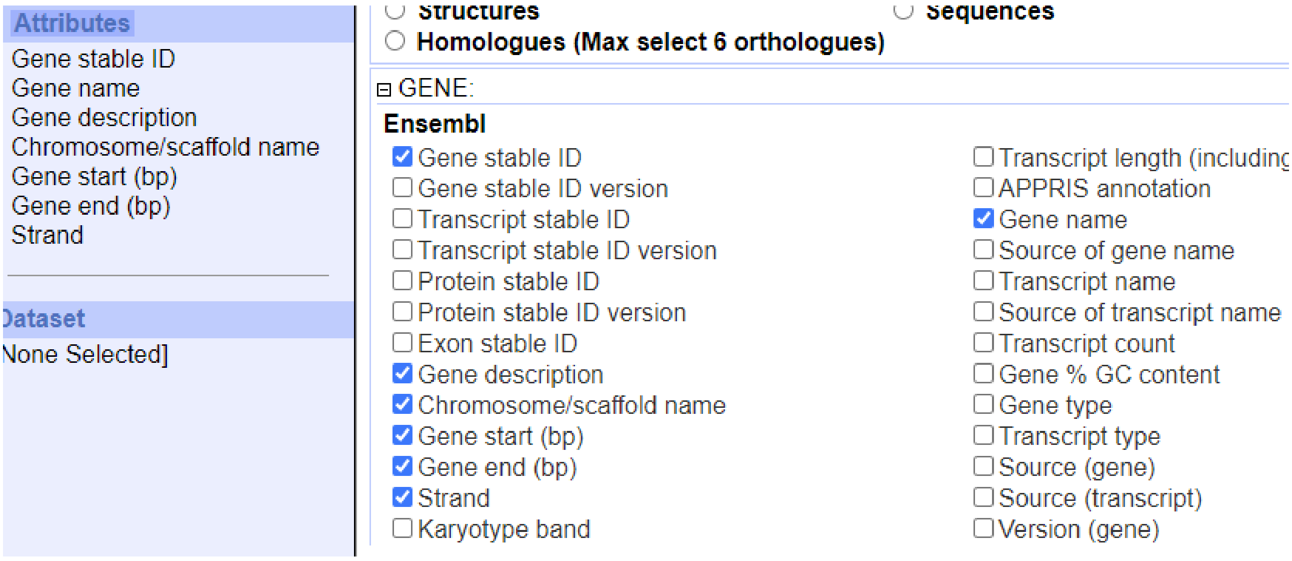
Go to https://www.ensembl.org/biomart/martview/afe982e758c87b06e672bc93a42a4f30.
Export all results to csv file.
Select dataset Ensembl Gene 101.
Select Species (refseq) (ex Cow genes (ARS-UCD1.2)).
Go to Attributes, Gene and create the biomart file following this header (select in the right order).
Gene stable ID,Gene name,Gene description,Chromosome/scaffold name,Gene start (bp),Gene end (bp),Strand



Dataset
Filters
[None selected]
Attributes
Gene stable ID
Gene stable ID version
Transcript stable ID
Transcript stable ID version

Dataset
[None Selected]

Ensembl Genes 101
Cow genes (ARS-UCD1.2)



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

Dataset
[None Selected]

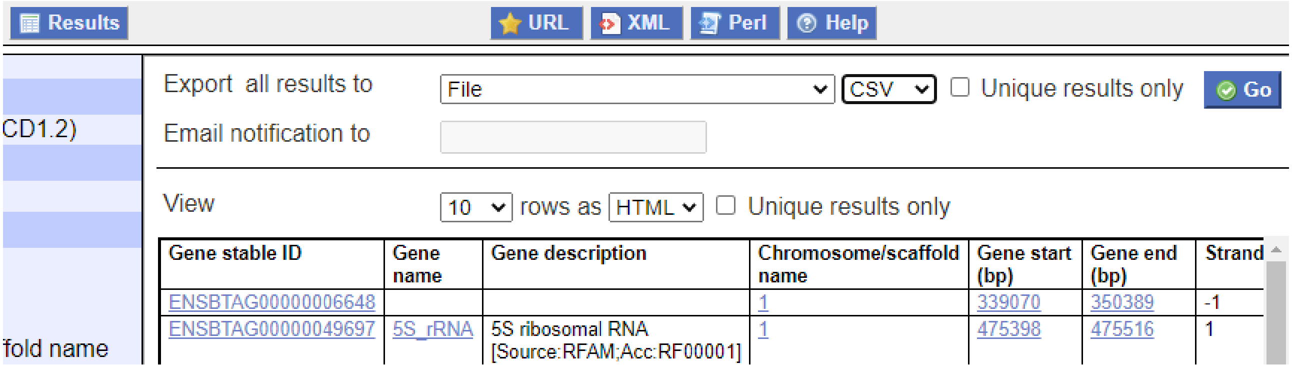
○ Structures ○ Sequences
○ Homologues (Max select 6 orthologues)

⊟ GENE:

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)



Export all results to File CSV □ Unique results only

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
ENSBTAG00000006648			1	339070	350389	-1
ENSBTAG00000049697	5S_rRNA	5S ribosomal RNA [Source:RFAM;Acc:RF00001]	1	475398	475516	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,tecnique,keyword_tags,relationship_note

2. PUBLICATION TABLE
   pubmed_ID,doi,article_title,article_authors,article_journal,publication_year

3. TECNIQUE TABLE
   tecnique,tecnique_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. KEGG TABLE
   kegg_ID,kegg_object_type,kegg_object_name

7. GEN_KEGG TABLE
   ensembl_gene_ID,kegg_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

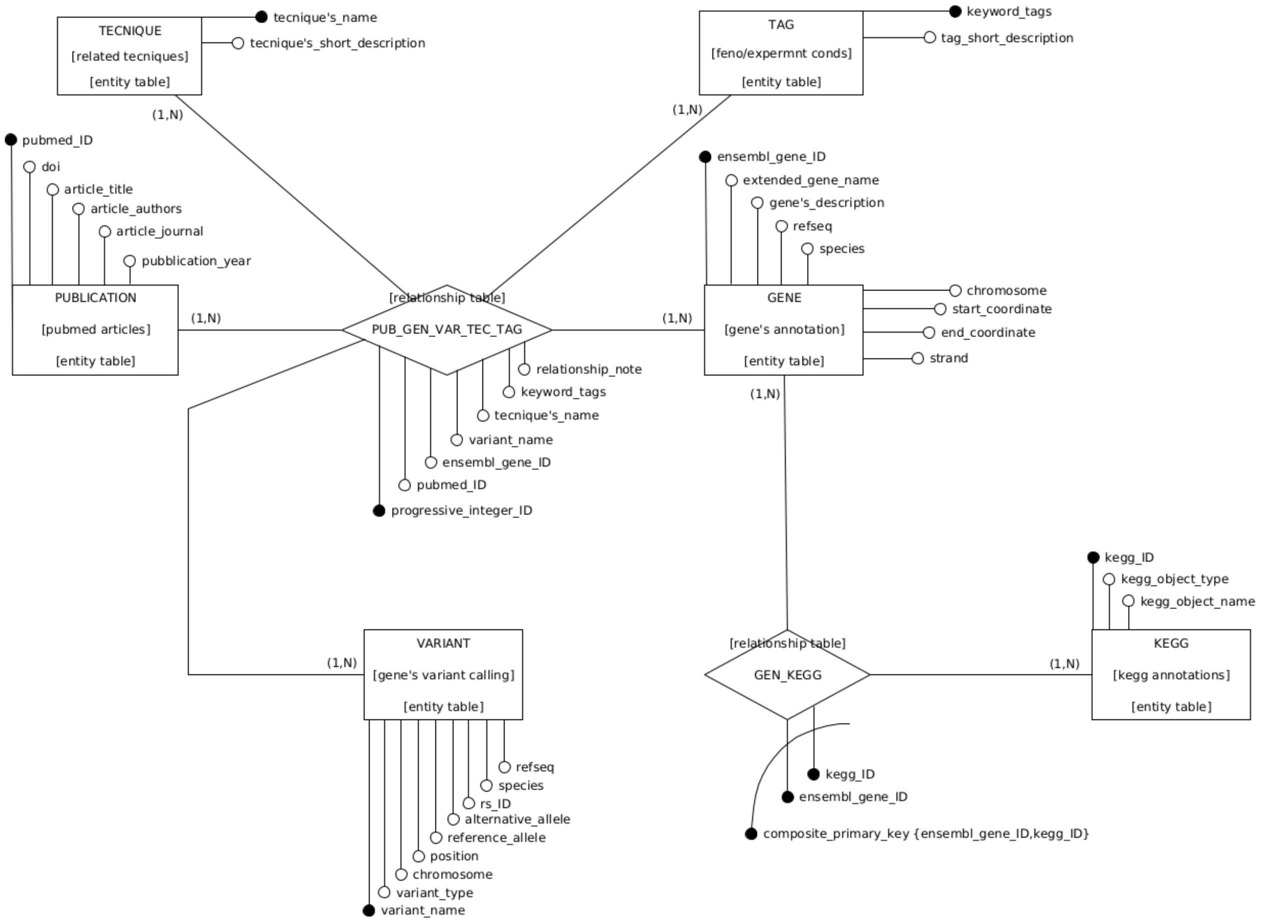
5) alternative allele (i.e. if the alternative allele is too long, try to find a way to summarize it)

example: hg38_1:146793_A|G

Chapter 7 Database Description

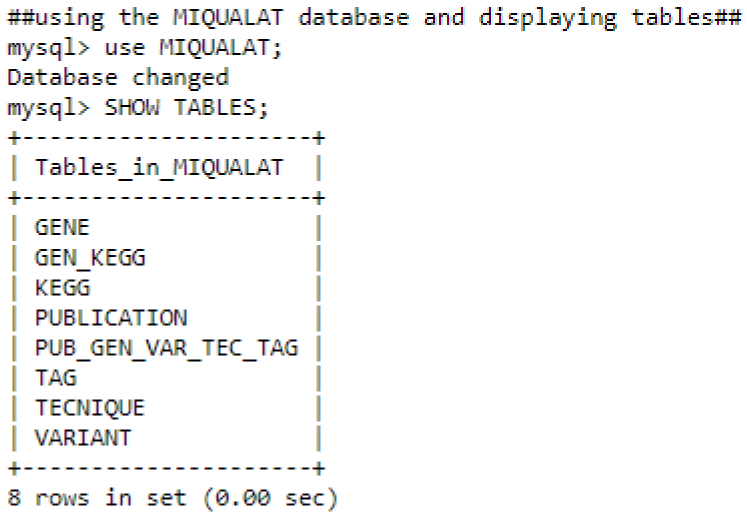
7.1 Database Schema

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

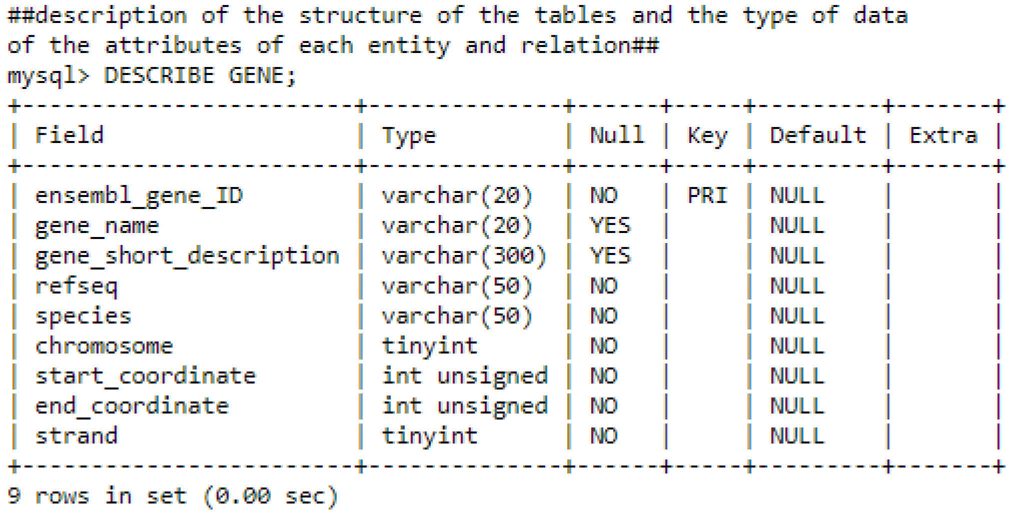


7.2 Database Description

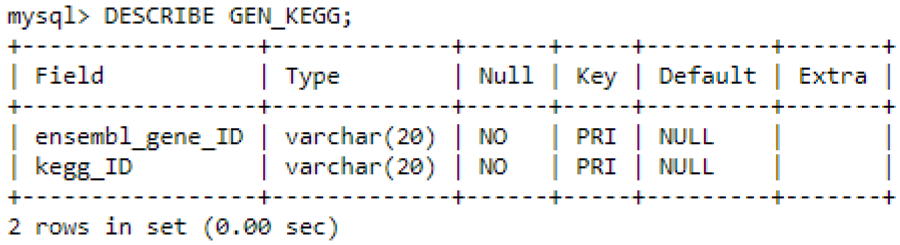
Below is an accurate description of all the tables in the database.



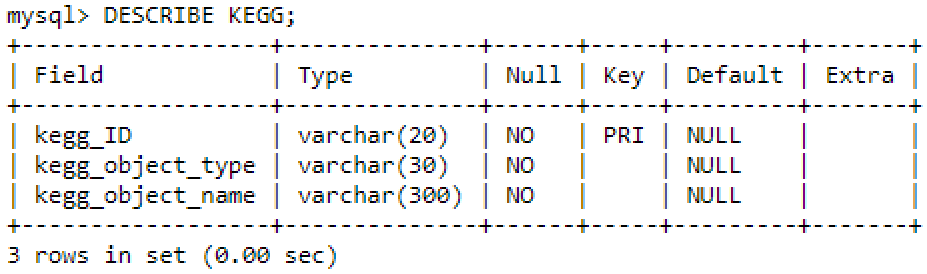
```
##using the MIQUALAT database and displaying tables##
mysql> use MIQUALAT;
Database changed
mysql> SHOW TABLES;
+----------------------+
| Tables_in_MIQUALAT   |
+----------------------+
| GENE                 |
| GEN_KEGG             |
| KEGG                 |
| PUBLICATION          |
| PUB_GEN_VAR_TEC_TAG  |
| TAG                  |
| TECNIQUE             |
| VARIANT              |
+----------------------+
8 rows in set (0.00 sec)
```



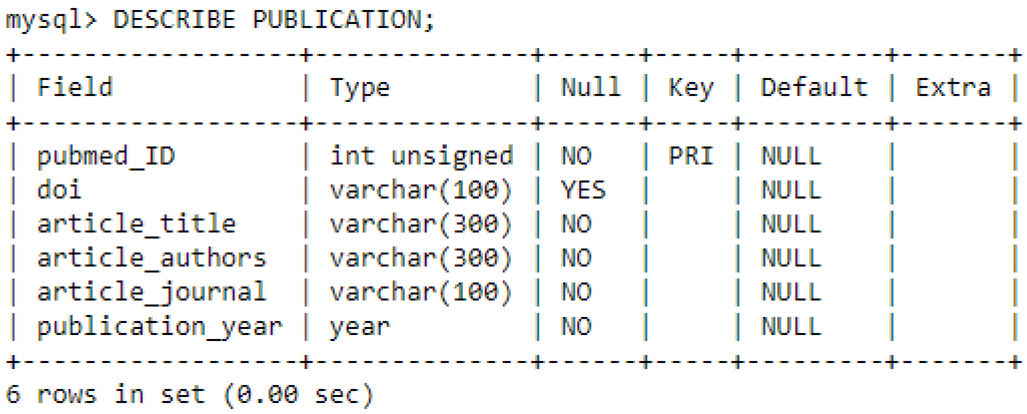
```
##description of the structure of the tables and the type of data
of the attributes of each entity and relation##
mysql> DESCRIBE GENE;
+-------------------------+--------------+------+-----+---------+-------+
| Field                   | Type         | Null | Key | Default | Extra |
+-------------------------+--------------+------+-----+---------+-------+
| 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)
```



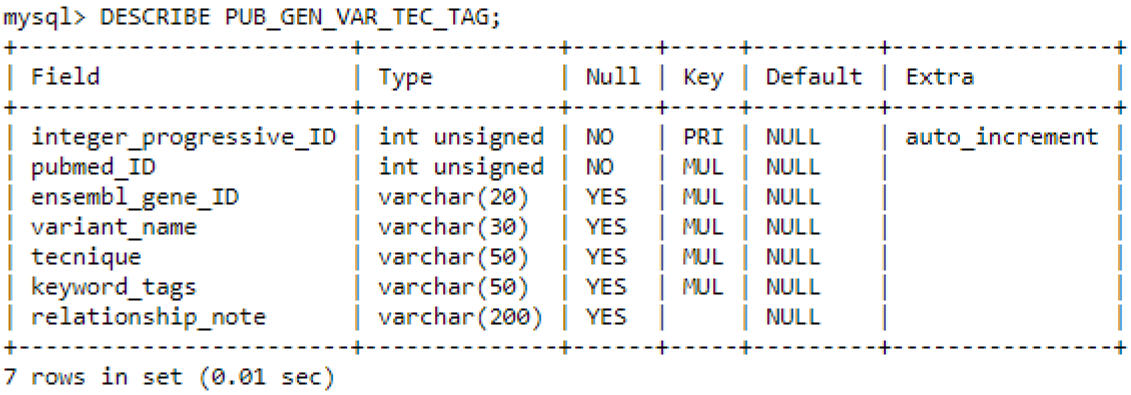
```
mysql> DESCRIBE GEN_KEGG;
+-----------------+-------------+------+-----+---------+-------+
| Field           | Type        | Null | Key | Default | Extra |
+-----------------+-------------+------+-----+---------+-------+
| ensembl_gene_ID | varchar(20) | NO   | PRI | NULL    |       |
| kegg_ID         | varchar(20) | NO   | PRI | NULL    |       |
+-----------------+-------------+------+-----+---------+-------+
2 rows in set (0.00 sec)
```



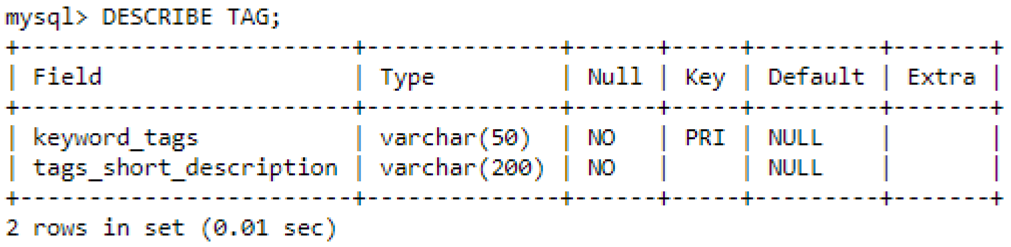
```
mysql> DESCRIBE KEGG;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| kegg_ID          | varchar(20)  | NO   | PRI | NULL    |       |
| kegg_object_type | varchar(30)  | NO   |     | NULL    |       |
| kegg_object_name | varchar(300) | NO   |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
3 rows in set (0.00 sec)
```



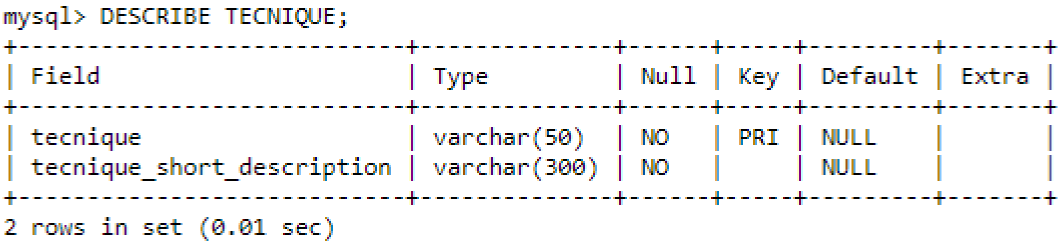
```
mysql> DESCRIBE PUBLICATION;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| pubmed_ID        | int unsigned | NO   | PRI | NULL    |       |
| 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)
```



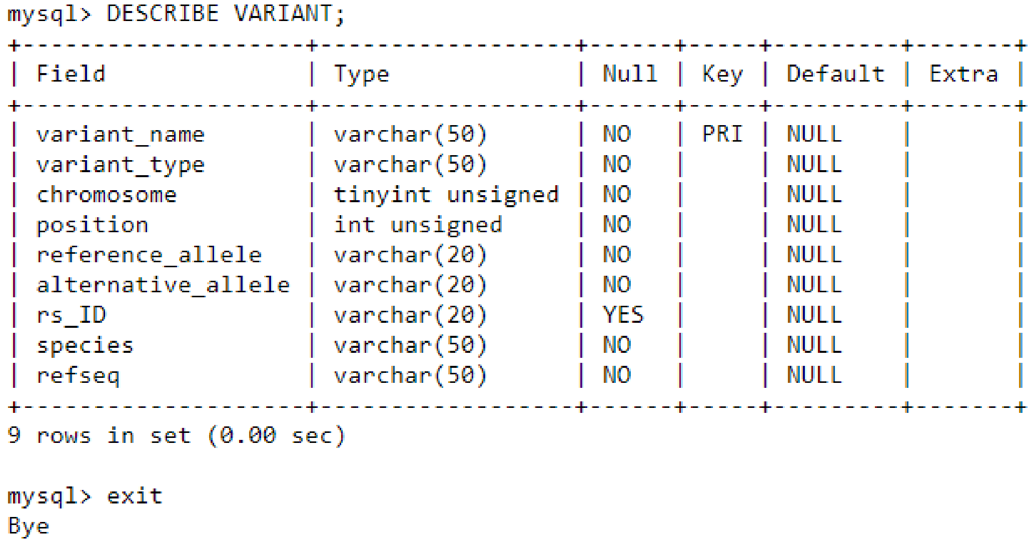
```
mysql> DESCRIBE PUB_GEN_VAR_TEC_TAG;
+------------------------+--------------+------+-----+---------+----------------+
| Field                  | Type         | Null | Key | Default | Extra          |
+------------------------+--------------+------+-----+---------+----------------+
| integer_progressive_ID | 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    |                |
| tecnique               | 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)
```

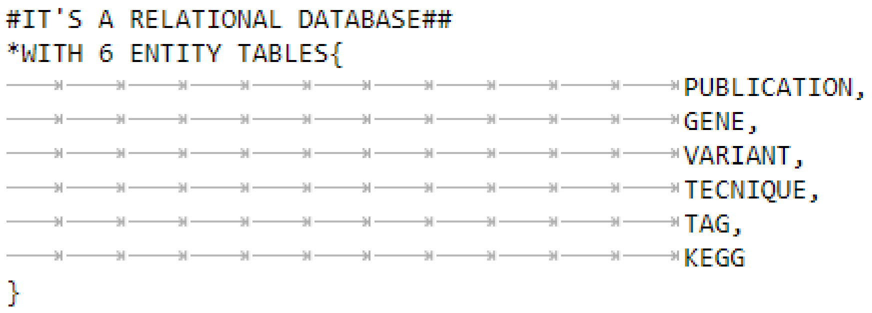


```
mysql> DESCRIBE TECNIQUE;
+----------------------------+--------------+------+-----+---------+-------+
| Field                      | Type         | Null | Key | Default | Extra |
+----------------------------+--------------+------+-----+---------+-------+
| tecnique                   | varchar(50)  | NO   | PRI | NULL    |       |
| tecnique_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
```



```
#IT'S A RELATIONAL DATABASE##
*WITH 6 ENTITY TABLES{
											PUBLICATION,
											GENE,
											VARIANT,
											TECNIQUE,
											TAG,
											KEGG
}
```

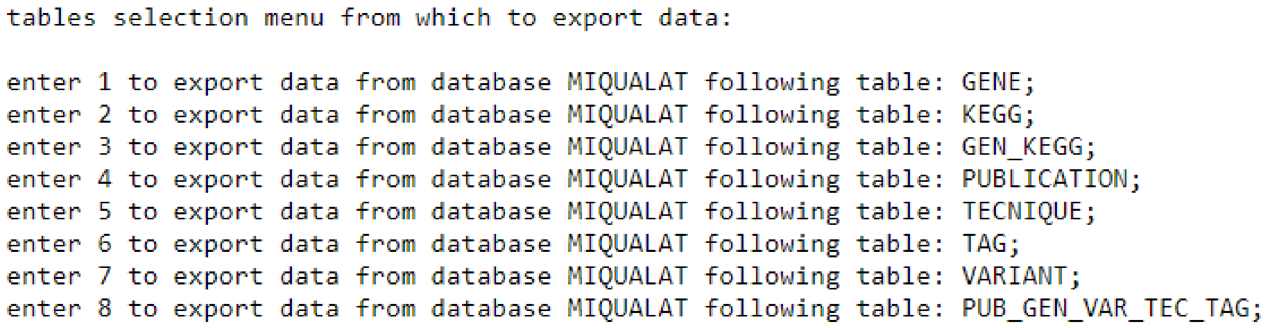
7.3 Checks Description

Below a short resume of database checks description.

1) fields number
2) duplicates lines
3) entries already present in the database
4) entry fields of a whole numerical nature
5) existence of foreign_keys for relational tables
6) special check for table for relational tables

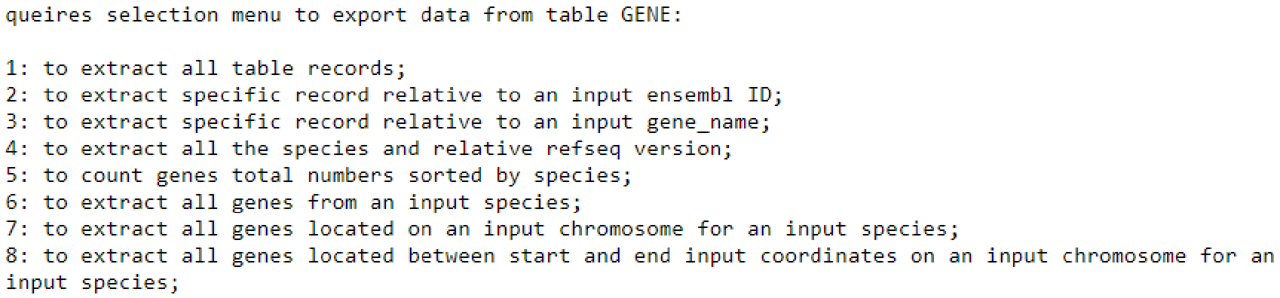
Chapter 8 Database Export Data

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.



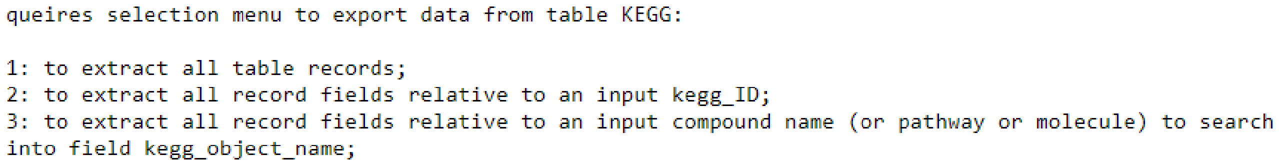
```
tables selection menu from which to export data:

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;
```



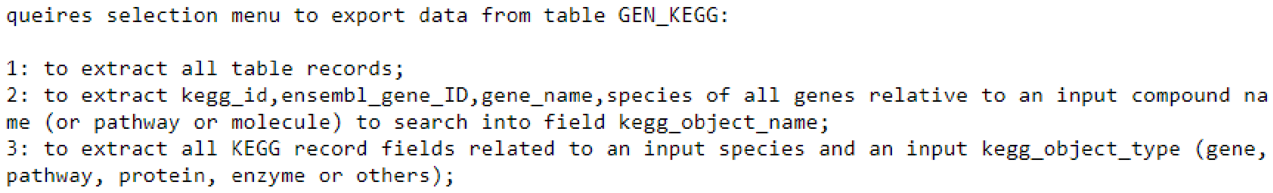
```
queires selection menu to export data from table GENE:

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;
```



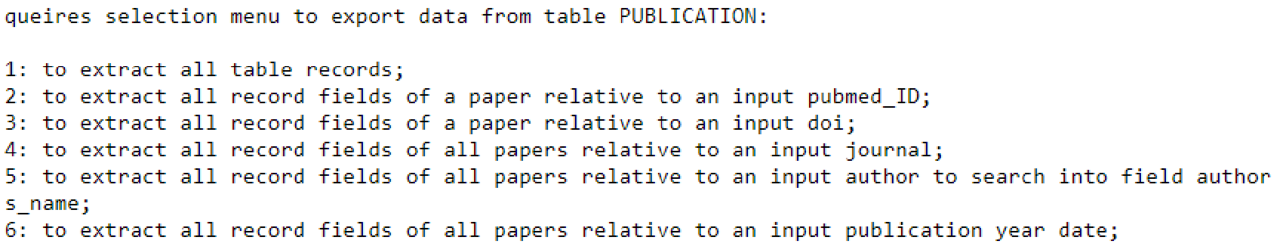
```
queires selection menu to export data from table KEGG:

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;
```



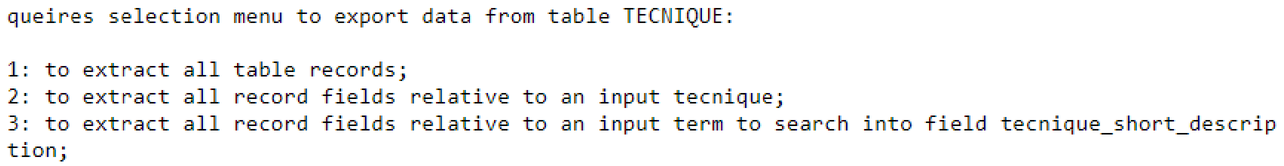
```
queires selection menu to export data from table GEN_KEGG:

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 na
me (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);
```



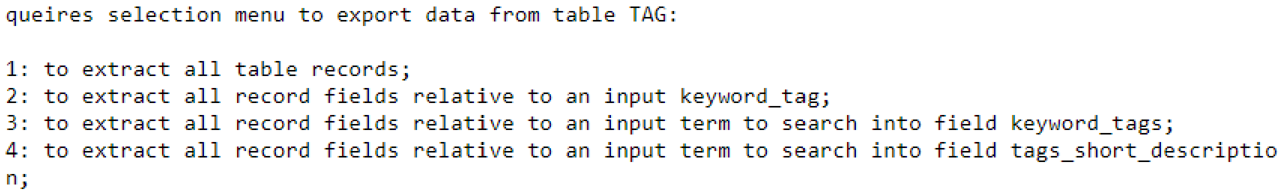
```
queires selection menu to export data from table PUBLICATION:

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;
```



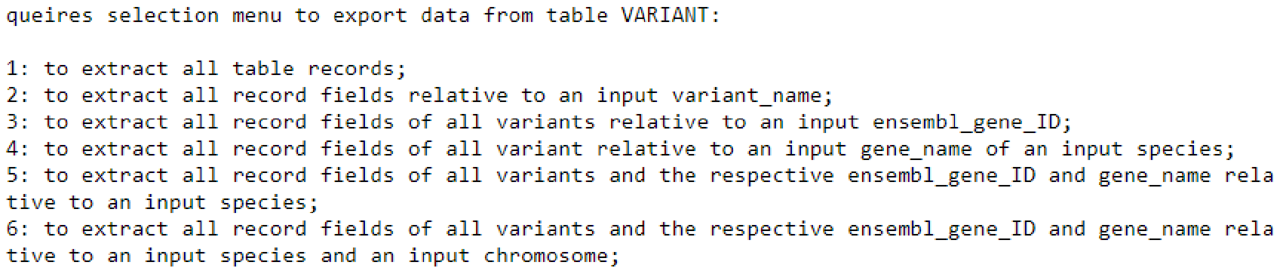
```
queires selection menu to export data from table TECNIQUE:

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_descrip
tion;
```



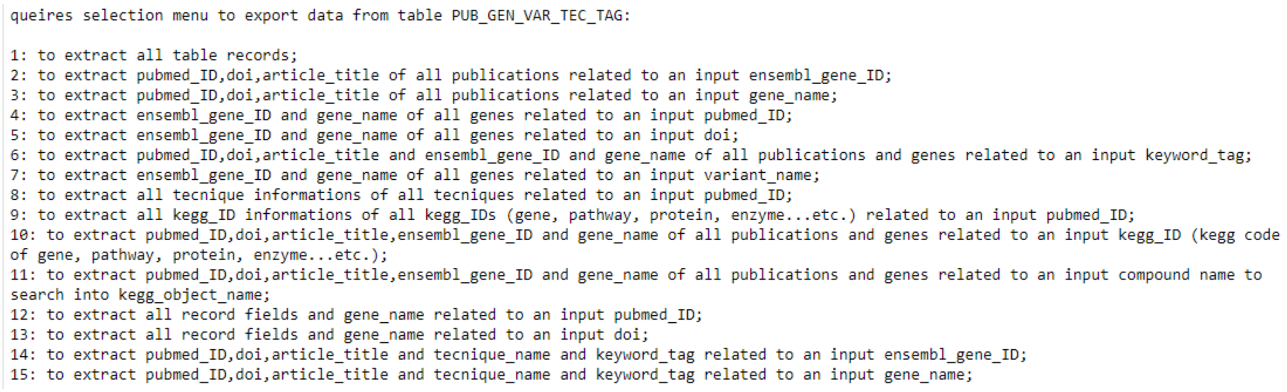
```
queires selection menu to export data from table TAG:

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_descriptio
n;
```



```
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 rela
tive to an input species;
6: to extract all record fields of all variants and the respective ensembl_gene_ID and gene_name rela
tive 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;
```