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, tecnique and other and associate them with descriptions and tags.

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.

This relational database allows you to link and export information from different tables.

crea
Consiglio per la ricerca in agricoltura
e l'analisi dell'economia agraria

UNIVERSITÀ DEGLI STUDI DELLA Tuscia
DIBAF
Dipartimento per la Innovazione nei sistemi
Biologici, Agroalimentari e Forestali

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 python3-pip.

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

Install jupyter notebook.

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

If you need to configure the configuration file is located in /root/.jupyter/jupyter_notebook_config.py

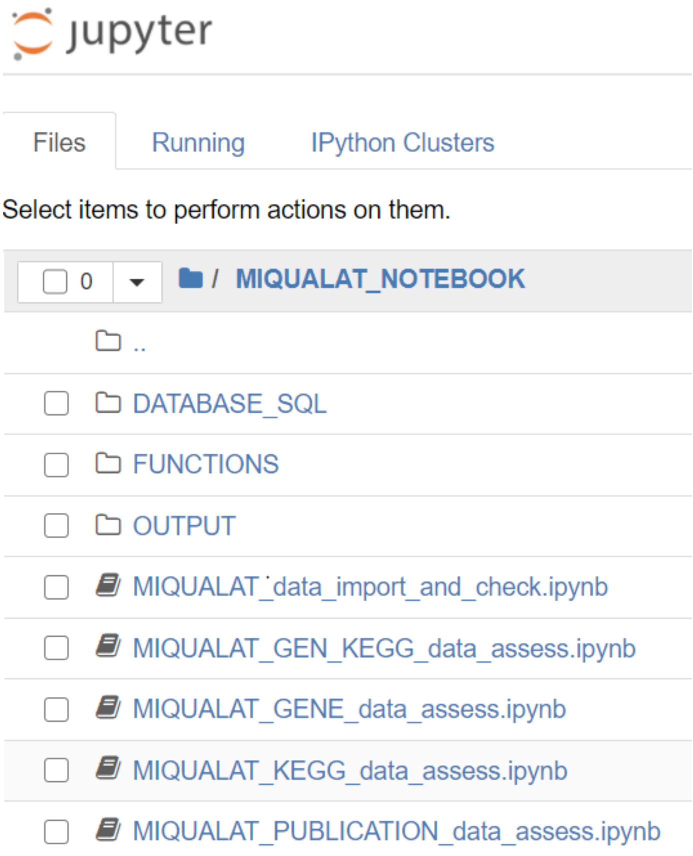
```
$ sudo jupyter-notebook --generate-config
```

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
```



Select items to perform actions on them.

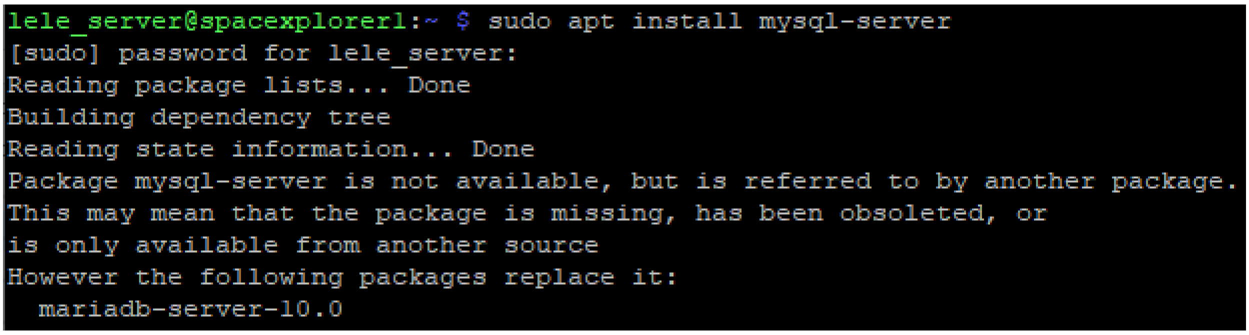
□ 0 ▾ / MIQUALAT_NOTEBOOK

- ..
- □ DATABASE_SQL
- □ FUNCTIONS
- □ OUTPUT
- □ MIQUALAT˙data_import_and_check.ipynb
- □ MIQUALAT_GEN_KEGG_data_assess.ipynb
- □ MIQUALAT_GENE_data_assess.ipynb
- □ MIQUALAT_KEGG_data_assess.ipynb
- □ MIQUALAT_PUBLICATION_data_assess.ipynb

2.2 Install Mysql Server

Install apache2 and mysql-server.
$ sudo apt install apache2
$ sudo apt install mysql-server

If you use raspberry or a similar system.
$ sudo apt install mariadb-server-10.0



```
lele_server@spacexplorer1:~ $ sudo apt install mysql-server
[sudo] password for lele_server:
Reading package lists... Done
Building dependency tree
Reading state information... Done
Package mysql-server is not available, but is referred to by another package.
This may mean that the package is missing, has been obsoleted, or
is only available from another source
However the following packages replace it:
  mariadb-server-10.0
```

Create user and set password for database (this user can be used 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';

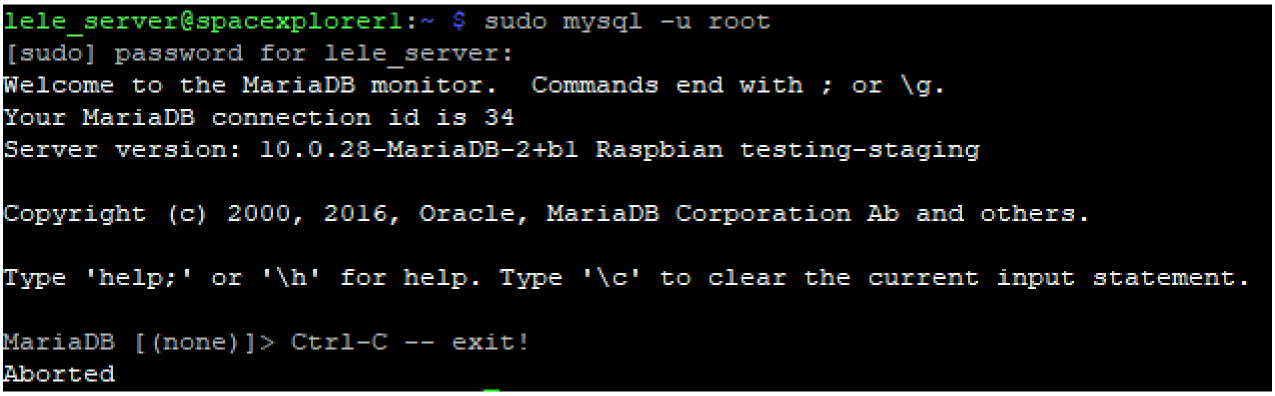
Grant privileges on one databases (safer).
$ GRANT ALL PRIVILEGES ON DATABASE. * TO 'user'@'localhost';
$ FLUSH PRIVILEGES;

!! IMPORTANT (pay attention before continuing with the command below), when you install phpmyadmin (command below) after select apache2 (with TAB) and press SPACE, 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)

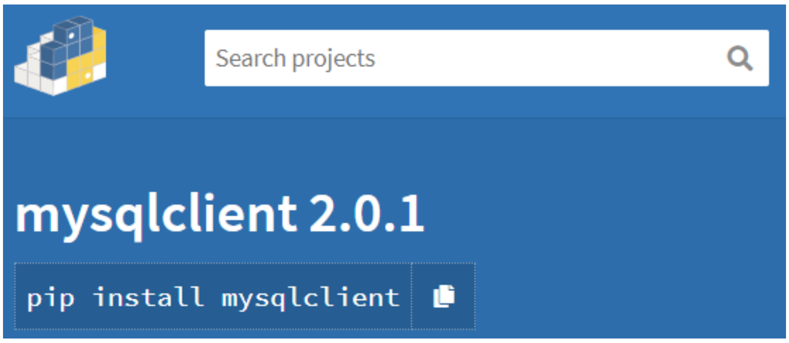


```
lele_server@spacexplorer1:~ $ sudo mysql -u root
[sudo] password for lele_server:
Welcome to the MariaDB monitor.  Commands end with ; or \g.
Your MariaDB connection id is 34
Server version: 10.0.28-MariaDB-2+b1 Raspbian testing-staging

Copyright (c) 2000, 2016, Oracle, MariaDB Corporation Ab and others.

Type 'help;' or '\h' for help. Type '\c' to clear the current input statement.

MariaDB [(none)]> Ctrl-C -- exit!
Aborted
```

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



mysqlclient 2.0.1

```
pip install mysqlclient
```

2.4 Install Required Python Libraries

The following libraries are required for use miqualat notebooks.
$ sudo pip3 install biopython
$ sudo pip3 install ensembl-rest
$ sudo pip3 install mygene

Chapter 3 Folder and Content Description

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.

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).

3. MIQUALAT_data_import_and_check.py
   Import and check the csv files processed with previous notebooks, using MySQLdb (in mysqlclient library).

4. pubmed_search.py
   Carry out searches within the pubmed database using biopython.

5. python_parser_biomart_gene_csv.py
   Convert the biomart file (which contains all biomart genes) into the right format for the GENE table.

☐ 0 ▾ / MIQUALAT_NOTEBOOK / FUNCTIONS

..

☐ ensembl_search.py

☐ ensembl_to_kegg_id.py

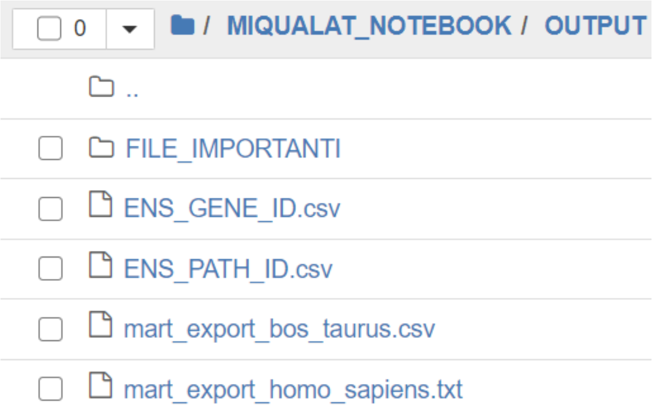
☐ MIQUALAT_data_import_and_check.py

☐ pubmed_search.py

☐ python_parser_biomart_gene_csv.py

Description of the OUTPUT Folder

In the OUTPUT folder are saved all the files processed by the notebook, to be imported into the miqualat database.



□ 0 / MIQUALAT_NOTEBOOK / OUTPUT

..

□ FILE_IMPORTANTI

□ ENS_GENE_ID.csv

□ ENS_PATH_ID.csv

□ mart_export_bos_taurus.csv

□ mart_export_homo_sapiens.txt

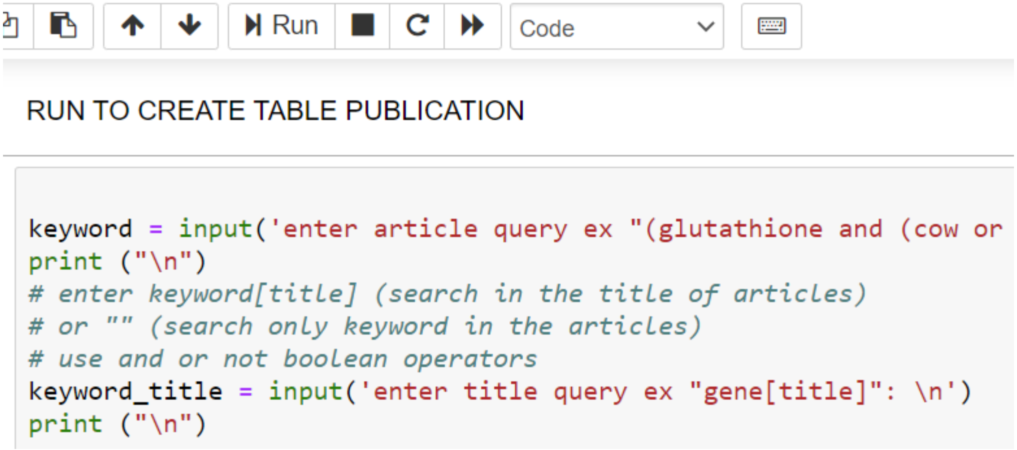
Description of the DATABASE Folder

In the OUTPUT folder are saved all the files processed by the notebook, to be imported into the miqualat database

Chapter 4 Miqualat Notebooks Description and Usage

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.

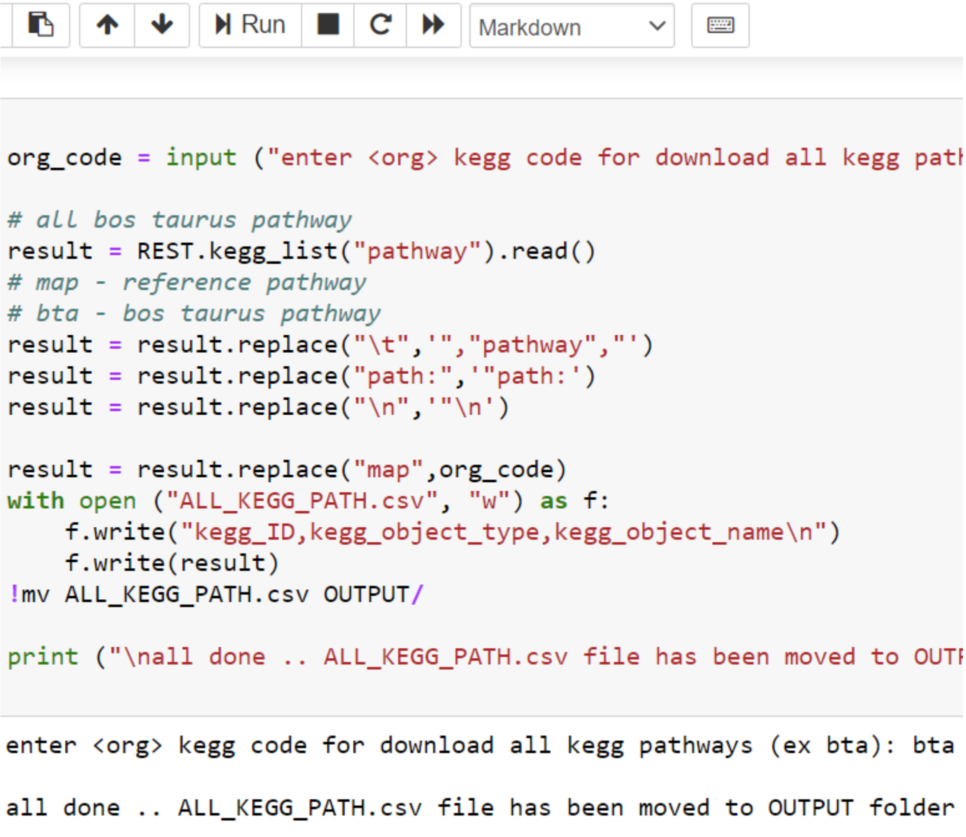
1. MIQUALAT_PUBLICATION_data_assess.ipynb
   Carry out searches within the pubmed database, automatize data collection for PUBLICAITON table.
   Creates PUBLICATION.csv file and move all searches (PUBMED_SEARCH_DATA_Y-m-d_H-M-S.txt format) in the OUTPUT folder.



RUN TO CREATE TABLE PUBLICATION

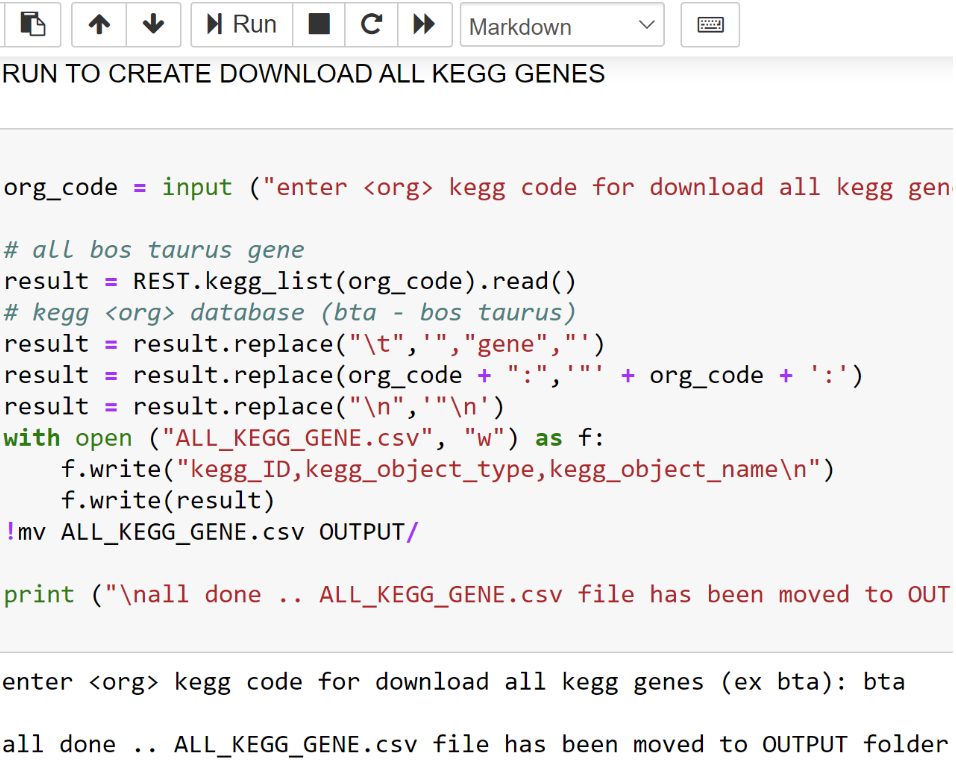
```
keyword = input('enter article query ex "(glutathione and (cow or
print ("\n")
# enter keyword[title] (search in the title of articles)
# or "" (search only keyword in the articles)
# use and or not boolean operators
keyword_title = input('enter title query ex "gene[title]": \n')
print ("\n")
```

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.



```
org_code = input ("enter <org> kegg code for download all kegg path

# all bos taurus pathway
result = REST.kegg_list("pathway").read()
# map - reference pathway
# bta - bos taurus pathway
result = result.replace("\t",'","pathway","')
result = result.replace("path:",'"path:')
result = result.replace("\n",'"\n')

result = result.replace("map",org_code)
with open ("ALL_KEGG_PATH.csv", "w") as f:
    f.write("kegg_ID,kegg_object_type,kegg_object_name\n")
    f.write(result)
!mv ALL_KEGG_PATH.csv OUTPUT/

print ("\nall done .. ALL_KEGG_PATH.csv file has been moved to OUTF
```

```
enter <org> kegg code for download all kegg pathways (ex bta): bta

all done .. ALL_KEGG_PATH.csv file has been moved to OUTPUT folder
```



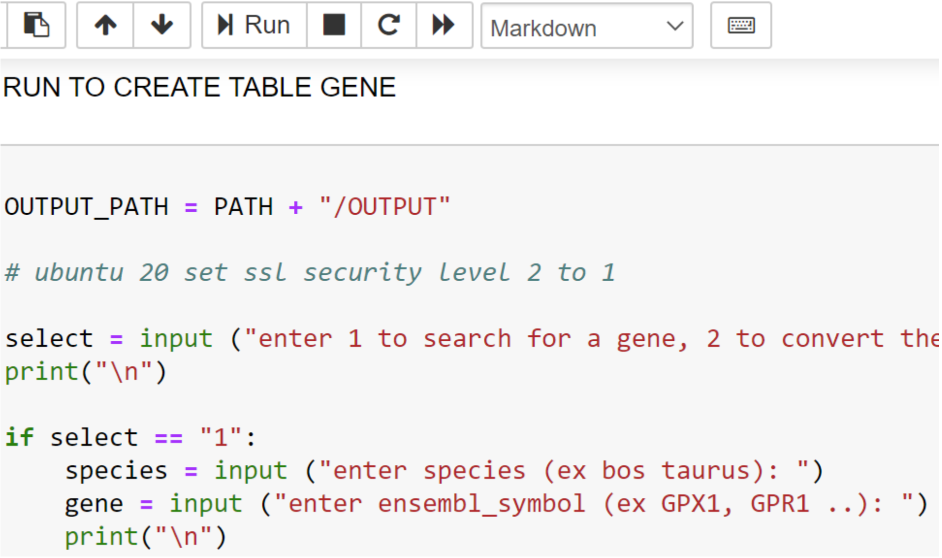
RUN TO CREATE DOWNLOAD ALL KEGG GENES

```
org_code = input ("enter <org> kegg code for download all kegg gen

# all bos taurus gene
result = REST.kegg_list(org_code).read()
# kegg <org> database (bta - bos taurus)
result = result.replace("\t",'","gene","')
result = result.replace(org_code + ":",'"' + org_code + ':')
result = result.replace("\n",'"\n')
with open ("ALL_KEGG_GENE.csv", "w") as f:
    f.write("kegg_ID,kegg_object_type,kegg_object_name\n")
    f.write(result)
!mv ALL_KEGG_GENE.csv OUTPUT/

print ("\nall done .. ALL_KEGG_GENE.csv file has been moved to OUT
```

```
enter <org> kegg code for download all kegg genes (ex bta): bta

all done .. ALL_KEGG_GENE.csv file has been moved to OUTPUT folder
```

3. MIQUALAT_GENE_data_assess.ipynb
   Allows you to download genes from ensembl database, automatize data collection for GENE table.

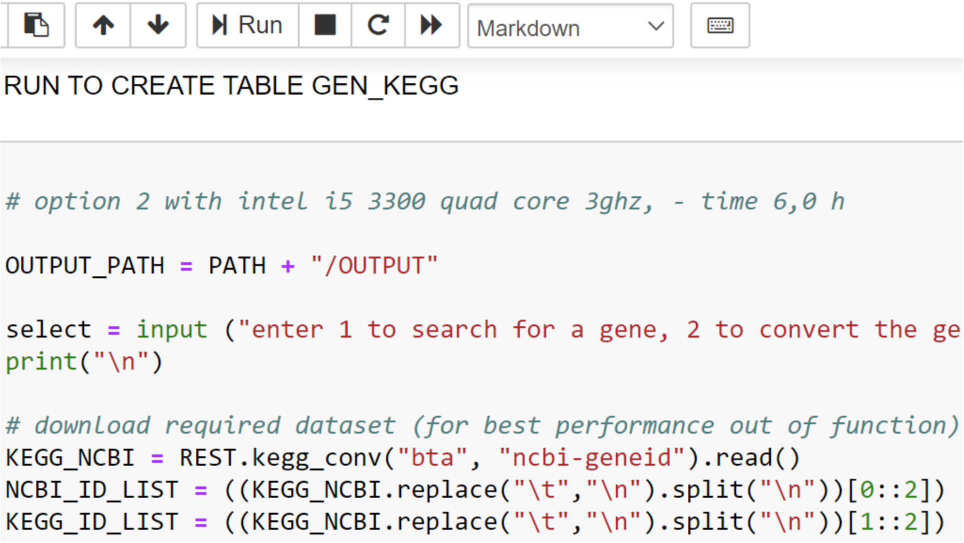
Option 1) get information from specific genes (ensembl gene id required).
Option 2) convert all genes downloaded from biomart (GENE.csv required).
In any case creates GENE.csv and move it in the OUTPUT folder.



RUN TO CREATE TABLE GENE

```
OUTPUT_PATH = PATH + "/OUTPUT"

# ubuntu 20 set ssl security level 2 to 1

select = input ("enter 1 to search for a gene, 2 to convert the
print("\n")

if select == "1":
    species = input ("enter species (ex bos taurus): ")
    gene = input ("enter ensembl_symbol (ex GPX1, GPR1 ..): ")
    print("\n")
```

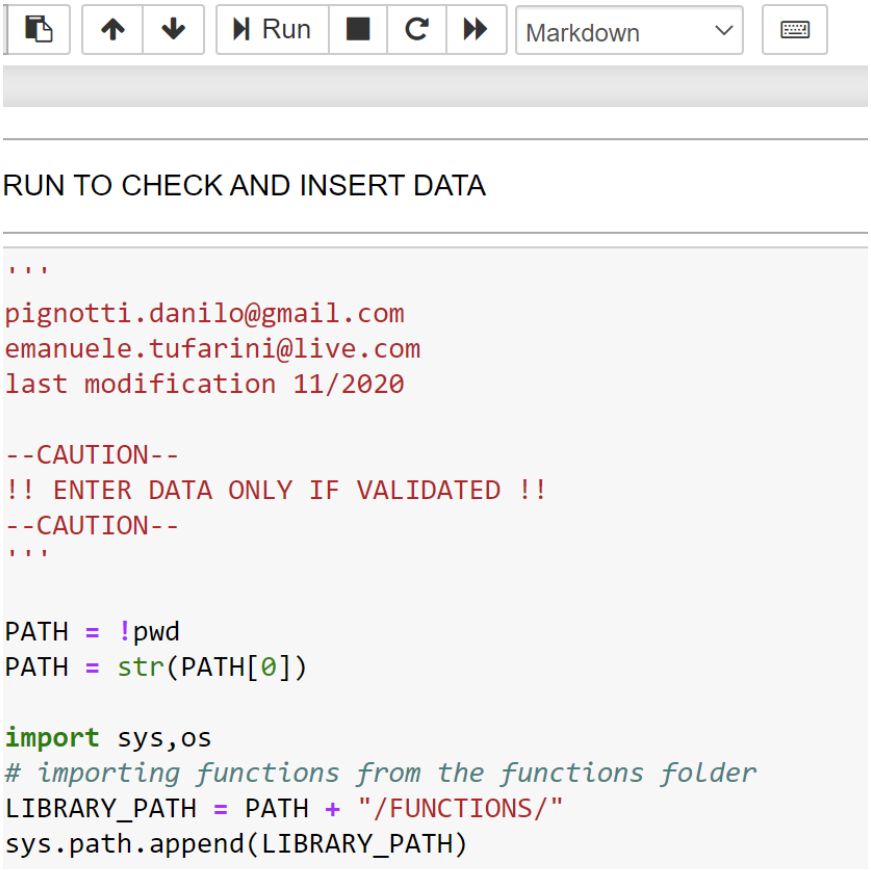
4. MIQUALAT_GEN_KEGG_data_assess.ipynb
Correlates the ensembl gene id to the kegg id (if possible) of the gene and related pathway, automatize data collection for GEN_KEGG table.
Option 1) correlates the ensembl gene id to the kegg id of the gene and related pathways for specific gene (ensembl gene id 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.



RUN TO CREATE TABLE GEN_KEGG

```
# option 2 with intel i5 3300 quad core 3ghz, - time 6,0 h

OUTPUT_PATH = PATH + "/OUTPUT"

select = input ("enter 1 to search for a gene, 2 to convert the ge
print("\n")

# download required dataset (for best performance out of function)
KEGG_NCBI = REST.kegg_conv("bta", "ncbi-geneid").read()
NCBI_ID_LIST = ((KEGG_NCBI.replace("\t","\n").split("\n"))[0::2])
KEGG_ID_LIST = ((KEGG_NCBI.replace("\t","\n").split("\n"))[1::2])
```

5. 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.



RUN TO CHECK AND INSERT DATA

```
'''
pignotti.danilo@gmail.com
emanuele.tufarini@live.com
last modification 11/2020

--CAUTION--
!! ENTER DATA ONLY IF VALIDATED !!
--CAUTION--
'''

PATH = !pwd
PATH = str(PATH[0])

import sys,os
# importing functions from the functions folder
LIBRARY_PATH = PATH + "/FUNCTIONS/"
sys.path.append(LIBRARY_PATH)
```

Chapter 5 Create Biomart File

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

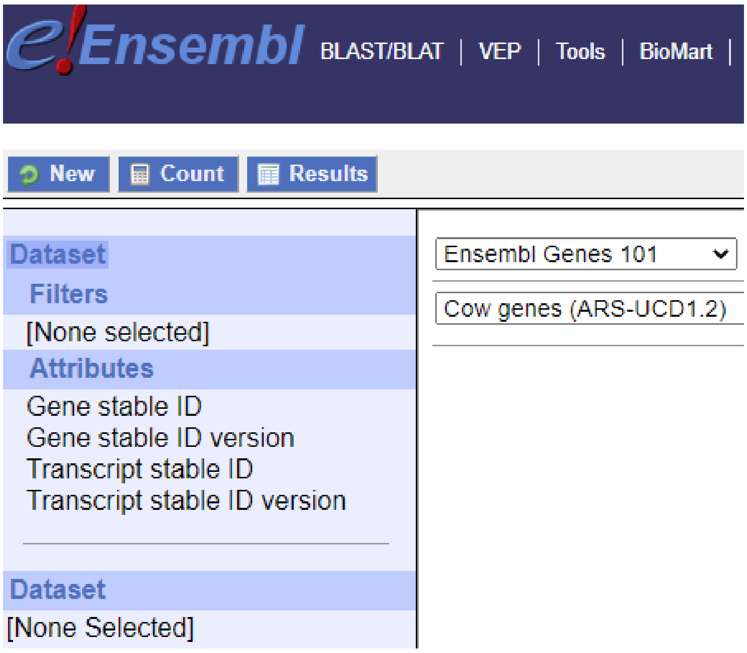
Export all results to File CSV.

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

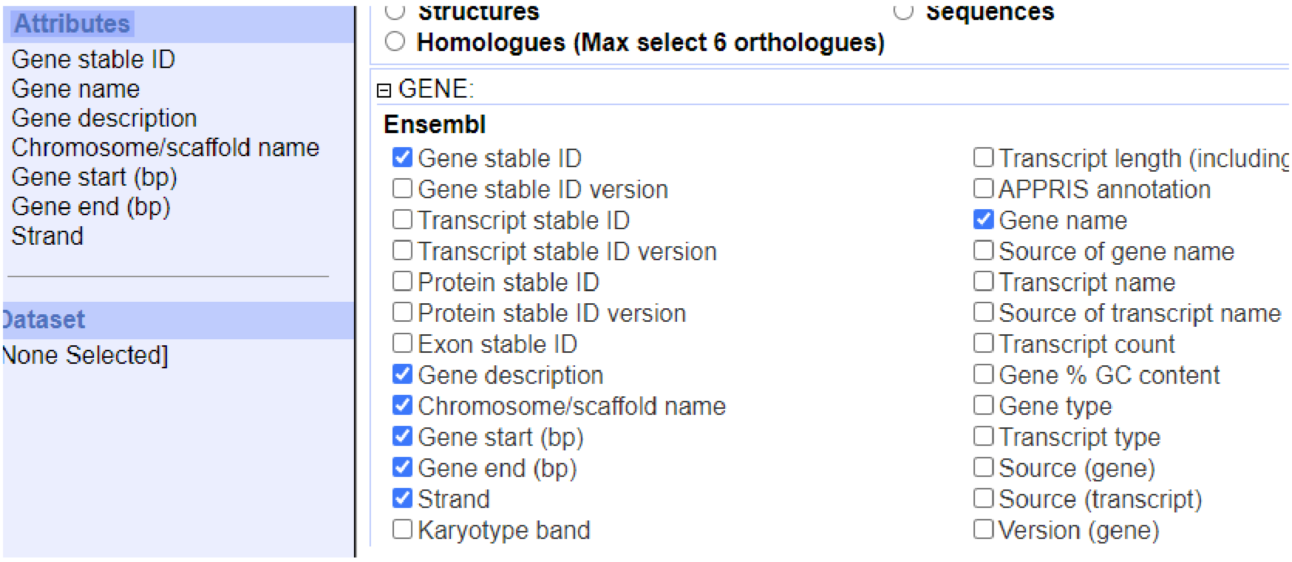
Gene stable ID
Gene stable ID version
Transcript stable ID
Transcript stable ID version

Dataset

[None Selected]

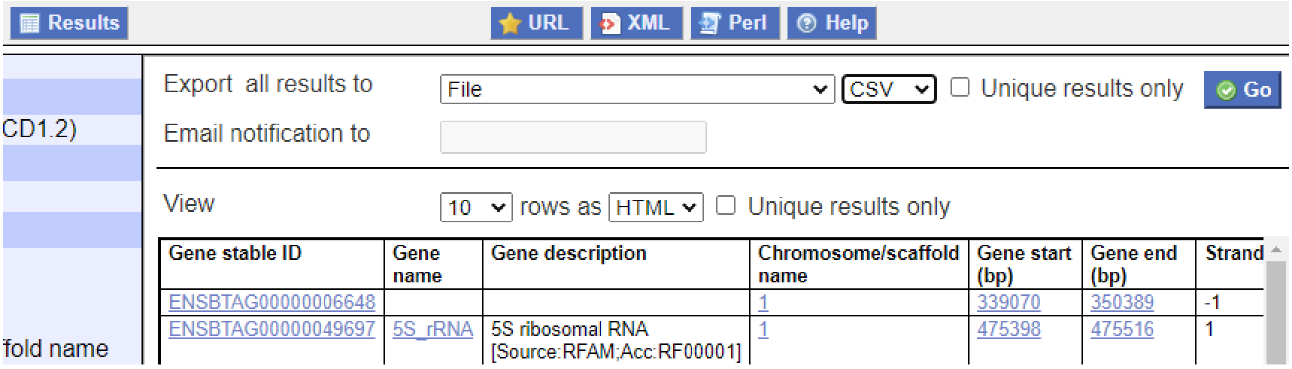
Ensembl Genes 101

Cow genes (ARS-UCD1.2)



☑ 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 Table File (Manually)

To pass the check the 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

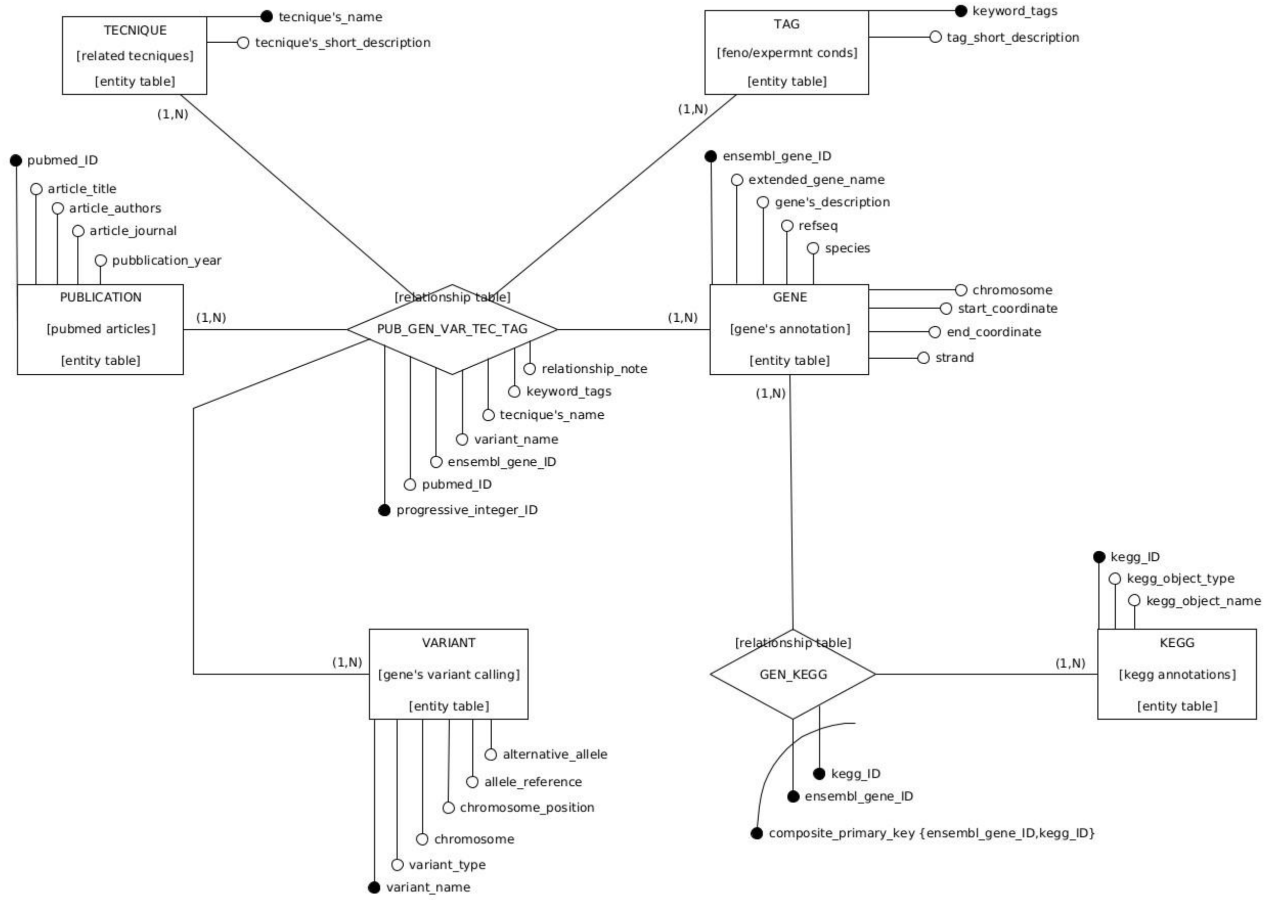
8. VARIANT TABLE

   variant_name,variant_type,chromosome,chromosome_position,allele_reference,alternative_allele_reference

Chapter 7 Database Description

7.1 Database Schema

Below the physical Entity-Relationship diagram (ERD) of the Miqualat database.



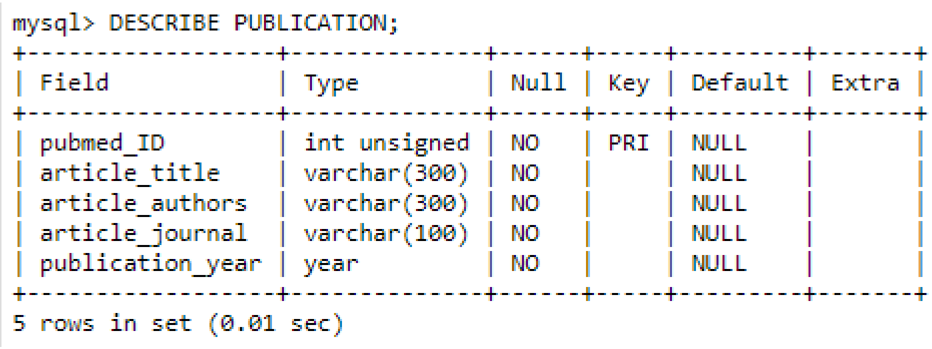
7.2 Database Description

```
##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)
```

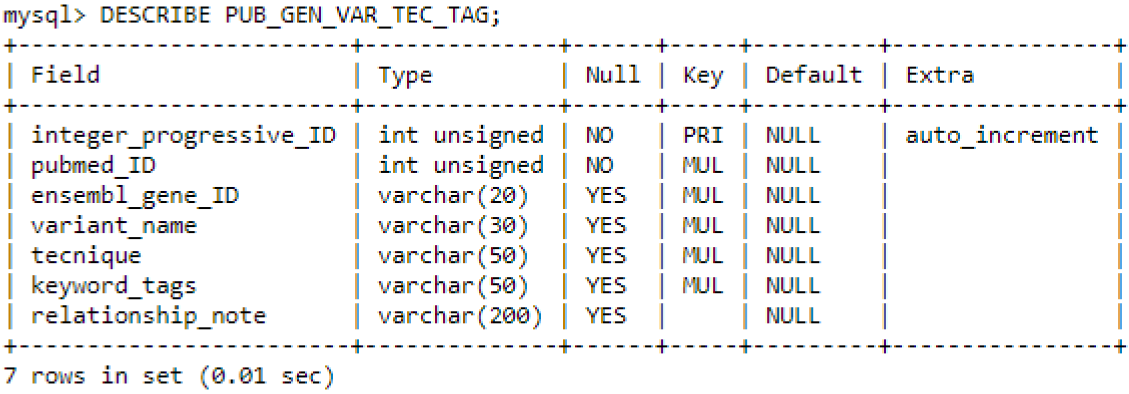
```
##description of the structure of the tables and the type of data
of the attributes of each entity and relation##
mysql> DESCRIBE GENE;
+-------------------------+--------------+------+-----+---------+-------+
| Field                   | Type         | Null | Key | Default | Extra |
+-------------------------+--------------+------+-----+---------+-------+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

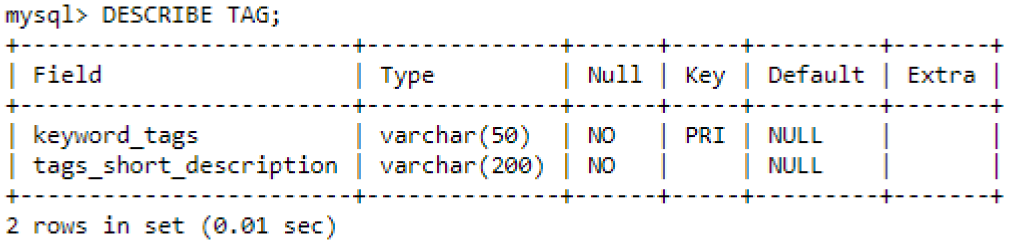
```
mysql> DESCRIBE KEGG;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| kegg_ID          | varchar(20)  | NO   | PRI | NULL    |       |
| kegg_object_type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



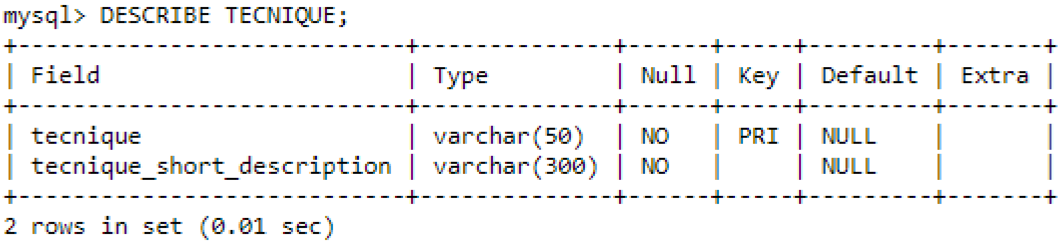
```
mysql> DESCRIBE PUBLICATION;
+------------------+--------------+------+-----+---------+-------+
| Field            | Type         | Null | Key | Default | Extra |
+------------------+--------------+------+-----+---------+-------+
| pubmed_ID        | int unsigned | NO   | PRI | NULL    |       |
| article_title    | varchar(300) | NO   |     | NULL    |       |
| article_authors  | varchar(300) | NO   |     | NULL    |       |
| article_journal  | varchar(100) | NO   |     | NULL    |       |
| publication_year | year         | NO   |     | NULL    |       |
+------------------+--------------+------+-----+---------+-------+
5 rows in set (0.01 sec)
```



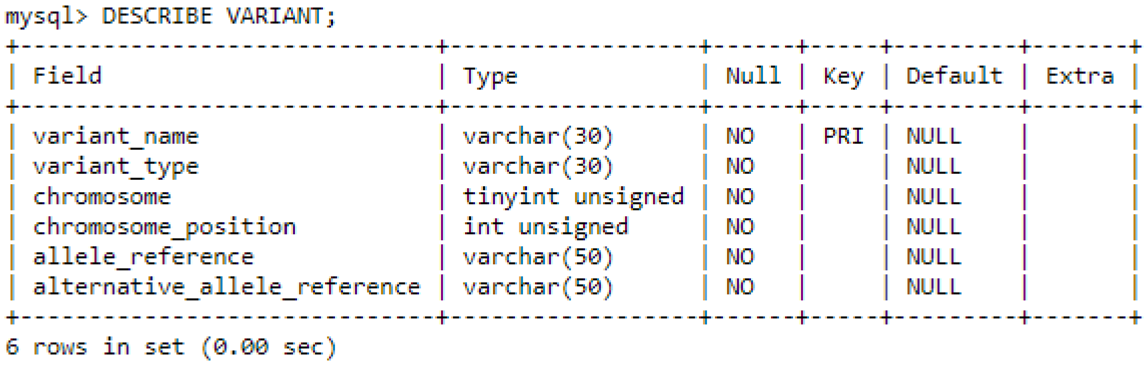
```
mysql> DESCRIBE PUB_GEN_VAR_TEC_TAG;
+------------------------+--------------+------+-----+---------+----------------+
| Field                  | Type         | Null | Key | Default | Extra          |
+------------------------+--------------+------+-----+---------+----------------+
| integer_progressive_ID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



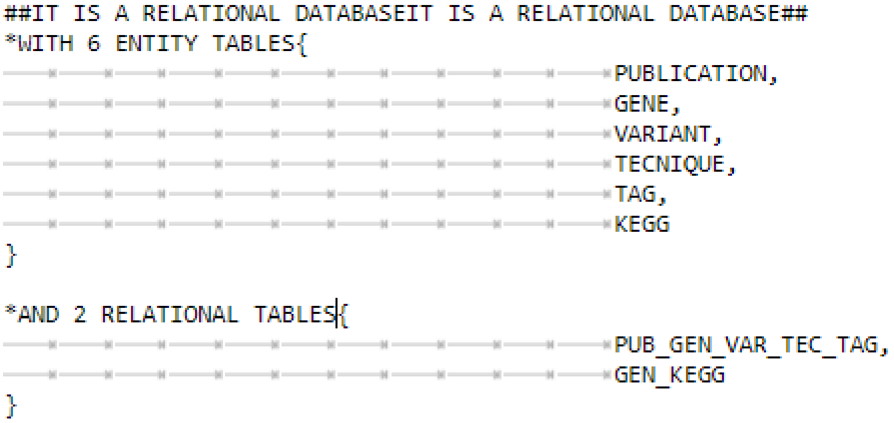
```
mysql> DESCRIBE TAG;
+------------------------+--------------+------+-----+---------+-------+
| Field                  | Type         | Null | Key | Default | Extra |
+------------------------+--------------+------+-----+---------+-------+
| keyword_tags           | varchar(50)  | NO   | PRI | NULL    |       |
| 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(30)      | NO   | PRI | NULL    |       |
| variant_type                 | varchar(30)      | NO   |     | NULL    |       |
| chromosome                   | tinyint unsigned | NO   |     | NULL    |       |
| chromosome_position          | int unsigned     | NO   |     | NULL    |       |
| allele_reference             | varchar(50)      | NO   |     | NULL    |       |
| alternative_allele_reference | varchar(50)      | NO   |     | NULL    |       |
+------------------------------+------------------+------+-----+---------+-------+
6 rows in set (0.00 sec)
```



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

*AND 2 RELATIONAL TABLES{
                                                PUB_GEN_VAR_TEC_TAG,
                                                GEN_KEGG
}
```

7.3 Check Description

1) entries fields number

2) entry duplicates

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 PUB_GEN_VAR_TEC_TAG

7) special check for table GEN_KEGG