Astroinformatics 2022
Session 06: Making and using database

Kinoshita Daisuke

24 October 2022
publicly accessible version

About this file...

- Important information about this file
  - The author of this file is Kinoshita Daisuke.
  - The original version of this file was used for the course "Astroinformatics" (course ID: AS6095) offered at Institute of Astronomy, National Central University from September 2022 to January 2023.
  - The file is provided in the hope that it will be useful, but there is no guarantee for the correctness. Use this file at your own risk.
  - If you are willing to use this file for your study, please feel free to use. I'll be very happy to receive feedback from you.
  - If you are willing to use this file for your teaching, please contact to Kinoshita Daisuke. When you use this file partly or entirely, please mention clearly that the author of the original version is Kinoshita Daisuke. Please help me to improve the contents of this file by sending your feedback.
  - Contact address: `https://www.instagram.com/daisuke23888/`

Nowadays, large volume of astronomical data obtained by both space-based and ground-based telescopes are available. Those data can be searched at data archive servers and be retrieved from them. To deal with flood of data, the knowledge about relational database management system and the query language "SQL" are essential. For this session, we try relational database management system.

1 Sample Python scripts for this session

Sample Python scripts for this session can be downloaded from GitHub repository. Visit following GitHub repository.

- `https://github.com/kinoshitadaisuke/ncu_astroinformatics_202209`

1.1 Executing sample Python scripts on a terminal emulator

If you prefer to execute sample Python scripts for this session on a terminal emulator, download `.py` files from GitHub repository.

1.2 Executing sample Python scripts on JupyterLab

If you prefer to execute sample Python scripts for this session on JupyterLab (or Jupyter Notebook), download `.ipynb` file from GitHub repository.

1.3 Executing sample Python scripts using Binder

If you prefer to execute sample Python scripts for this session on Binder, visit following web page.

- `https://mybinder.org/v2/gh/kinoshitadaisuke/ncu_astroinformatics_202209/HEAD`

Start your favourite web browser and go to above web page. (Fig. 1) In a minute or two, you see JupyterLab working on your web browser. (Fig. 2) Go to the directory (folder) "`s06`". (Fig. 3) Choose the file "`ai202209_s06.ipynb`" (Fig. 4 and 5) and open it (Fig. 6).

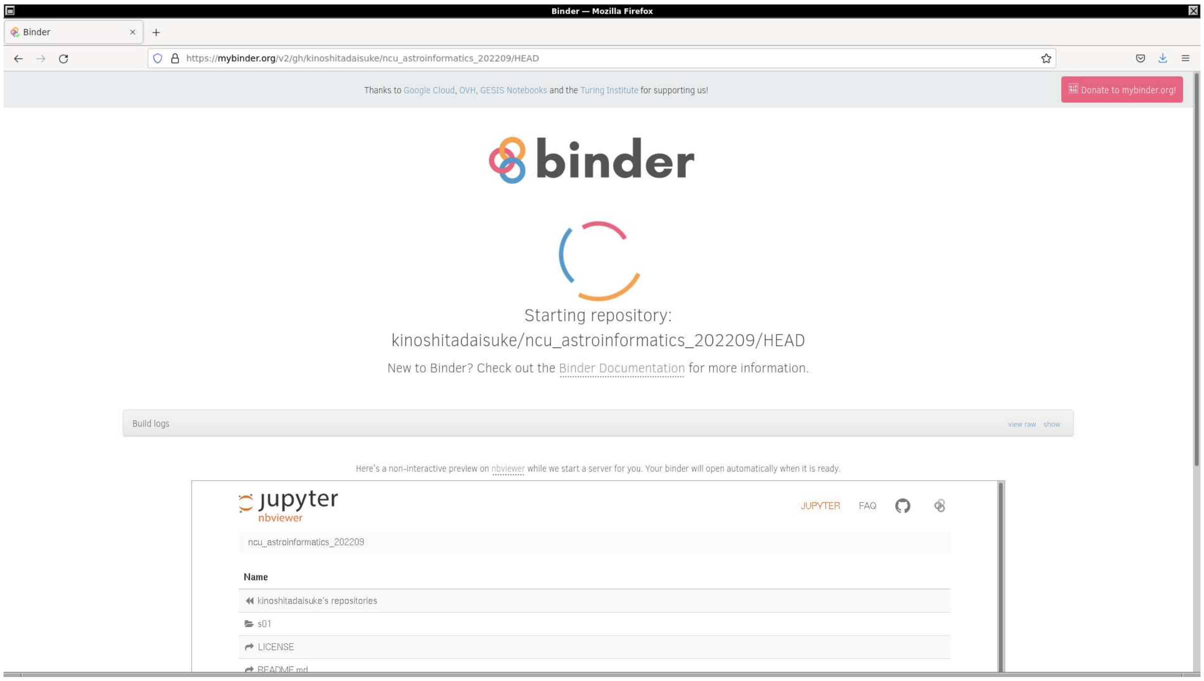


Figure 1: Using Binder to execute sample Python scripts for this session.

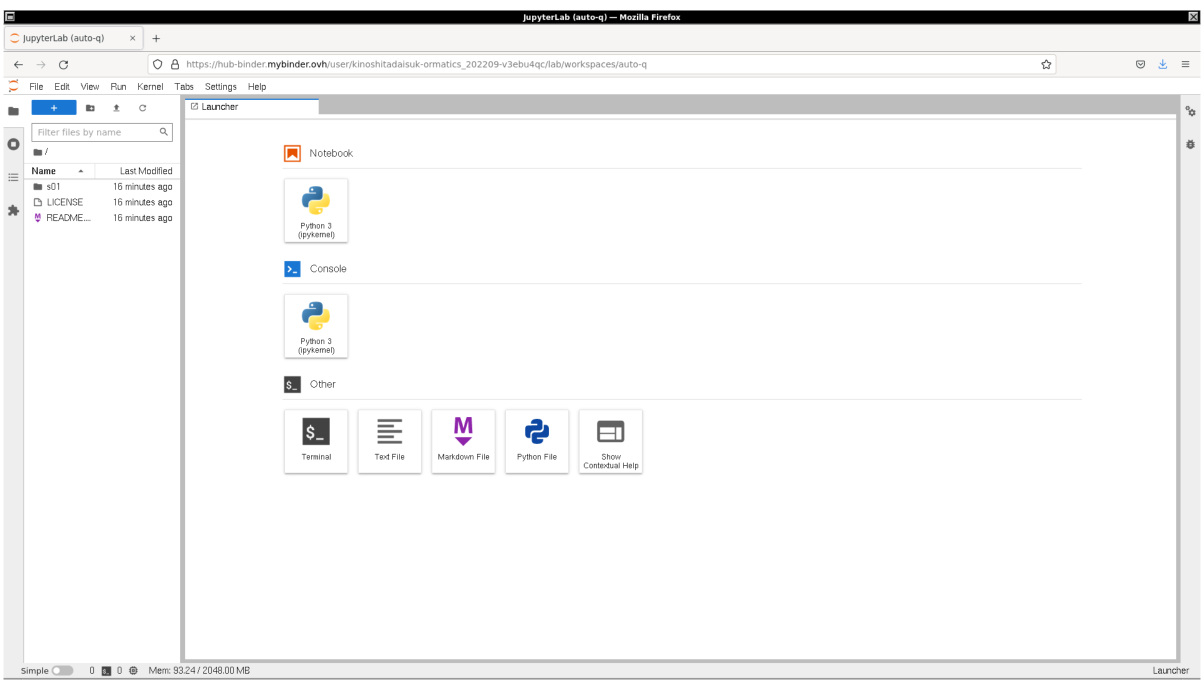


Figure 2: Using Binder to execute sample Python scripts for this session.

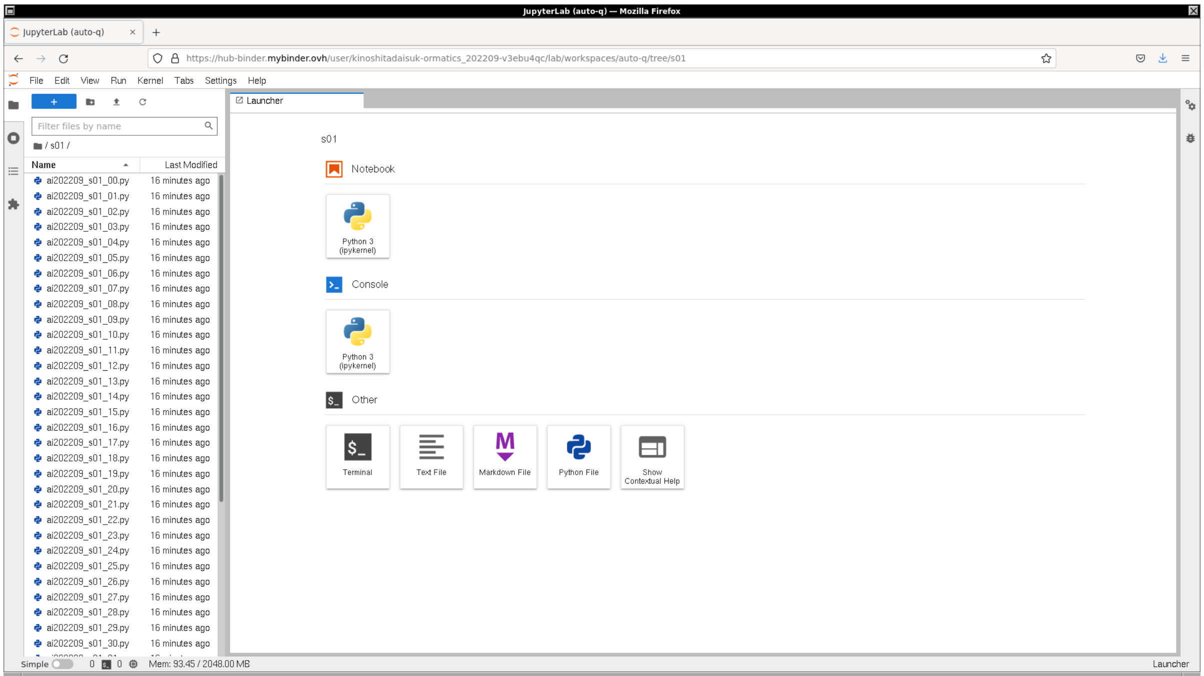


Figure 3: Using Binder to execute sample Python scripts for this session.

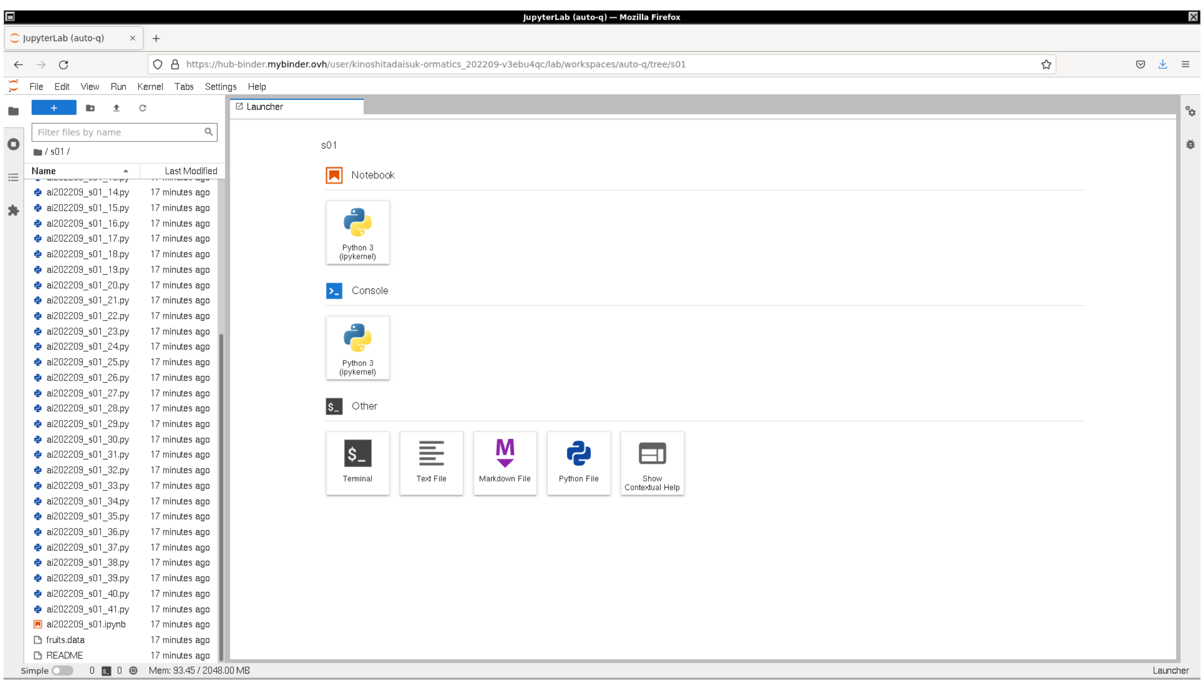


Figure 4: Using Binder to execute sample Python scripts for this session.

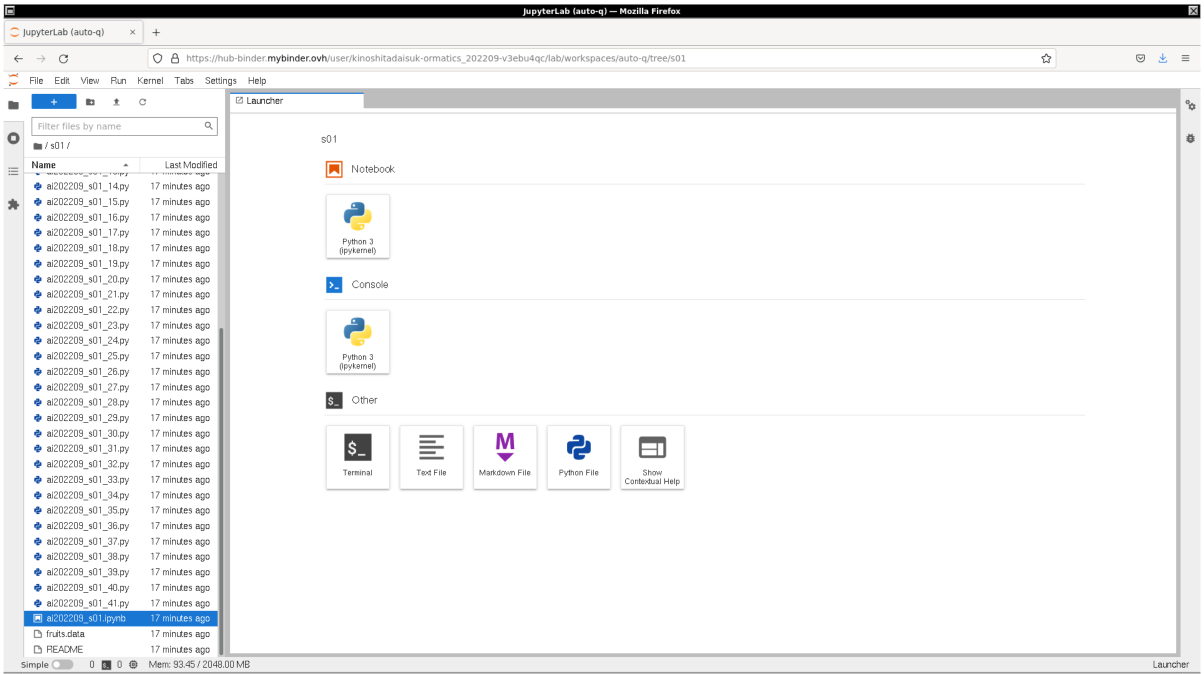


Figure 5: Using Binder to execute sample Python scripts for this session.

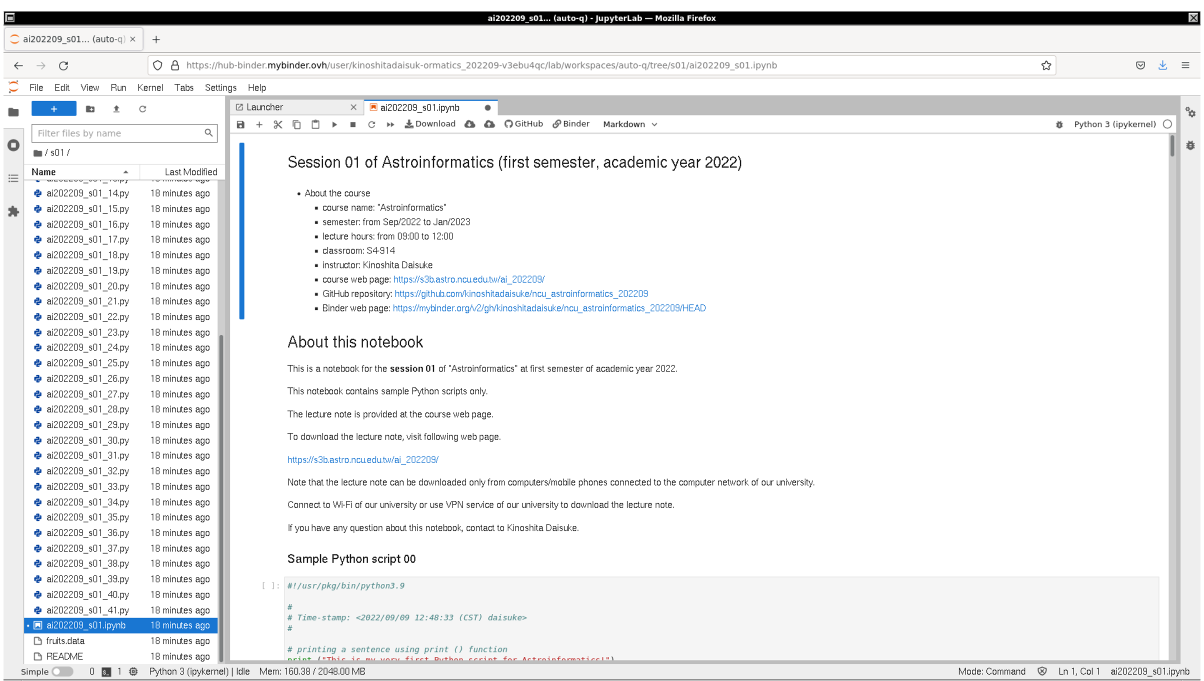


Figure 6: Using Binder to execute sample Python scripts for this session.

2 SQLite

For this session, we use the relational database management system "SQLite".

2.1 About SQLite

SQLite is a small, light-weight, and fast relational database engine. It is not a client-server type database management system, but it is a server-less database management system. We do not need to run the server process for use, and hence it is easy to use even for those who have not yet used the database management system. SQLite is a public domain relational database management system, and you can use it for free of charge.

To learn about SQLite, visit the official website of SQLite and read the documentation. The official website of SQLite can be found at following. (Fig. 7)

- SQLite: `https://www.sqlite.org/`
  - SQLite Documentation: `https://www.sqlite.org/docs.html`

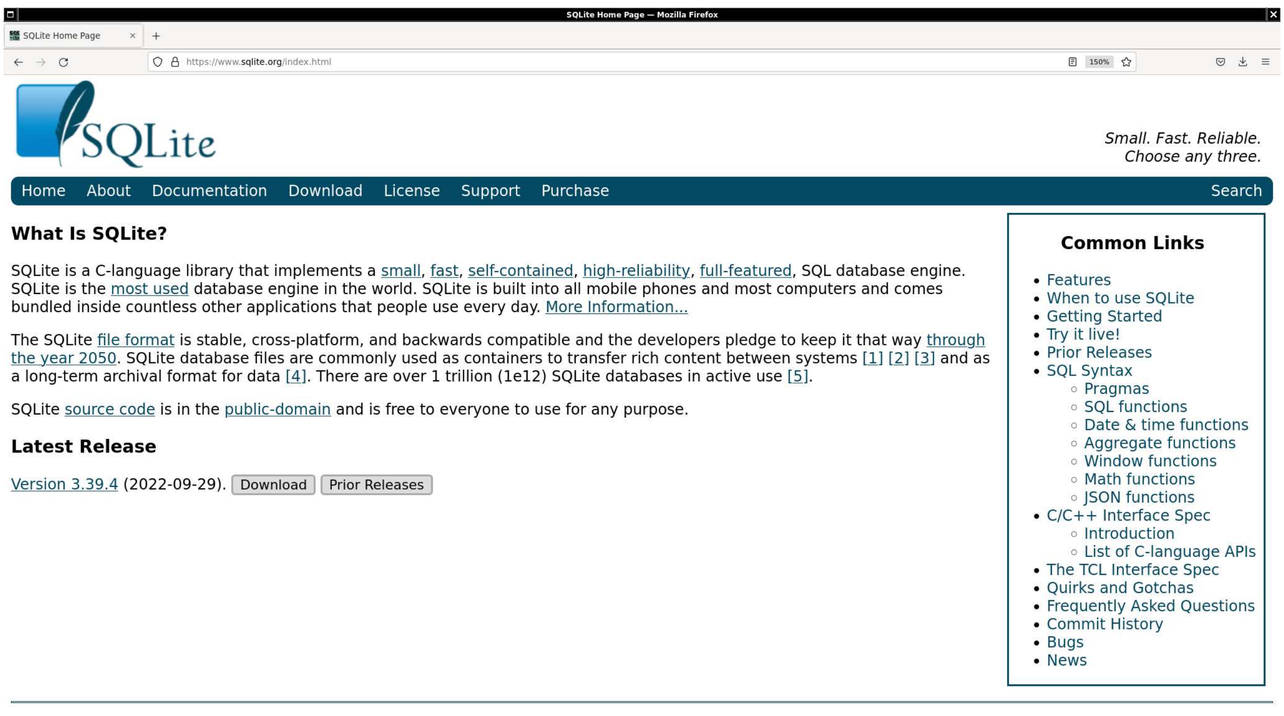


Figure 7: The official website of SQLite at `https://www.sqlite.org/`.

Our University has a licence to read following e-book. Read the book, and learn more about SQLite. (Fig. 8)

- "The Definitive Guide to SQLite", 2010, Grant Allen and Mike Owens, Apress, ISBN 978-1-4302-3226-1.
  - `https://link.springer.com/book/10.1007/978-1-4302-3226-1`

2.2 Using SQLite on a terminal emulator on your computer

If you prefer to use SQLite on a terminal emulator on your computer, try following command on a terminal emulator first.

```
% which sqlite3
/usr/pkg/bin/sqlite3
```

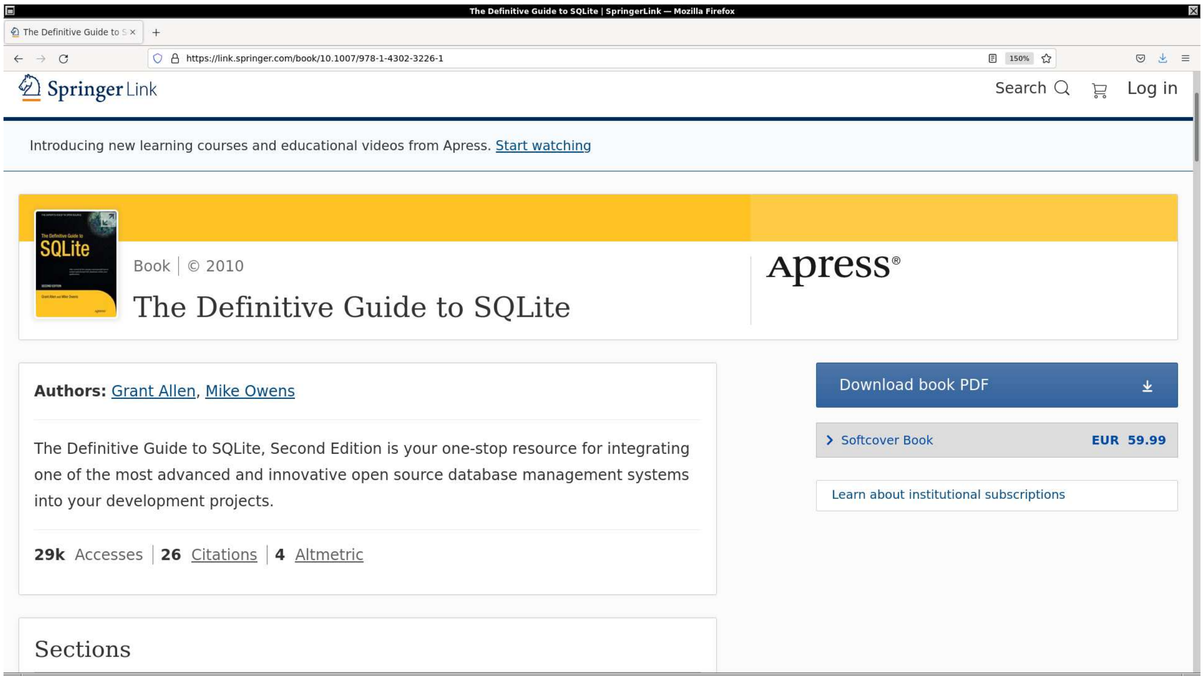


Figure 8: The official web page for the book "The Definitive Guide to SQLite" on the publisher's website.

If you have SQLite on your computer, the location of SQLite executable is shown.
If you do not have SQLite installed on your computer, you see following message.

```
% which sqlite3
sqlite3: Command not found.
```

In case you do not have SQLite on your computer,

- install SQLite on your own computer,
- or start your favourite web browser and use Binder.

If you have SQLite properly installed on your computer, try following command to start SQLite. (Fig. 9)

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

To quit SQLite, try `.quit` command. Note that SQLite commands start with a dot. (Fig. 10)

```
sqlite> .quit
```

Or, you may use `.exit` command. (Fig. 11)

```
sqlite> .exit
```

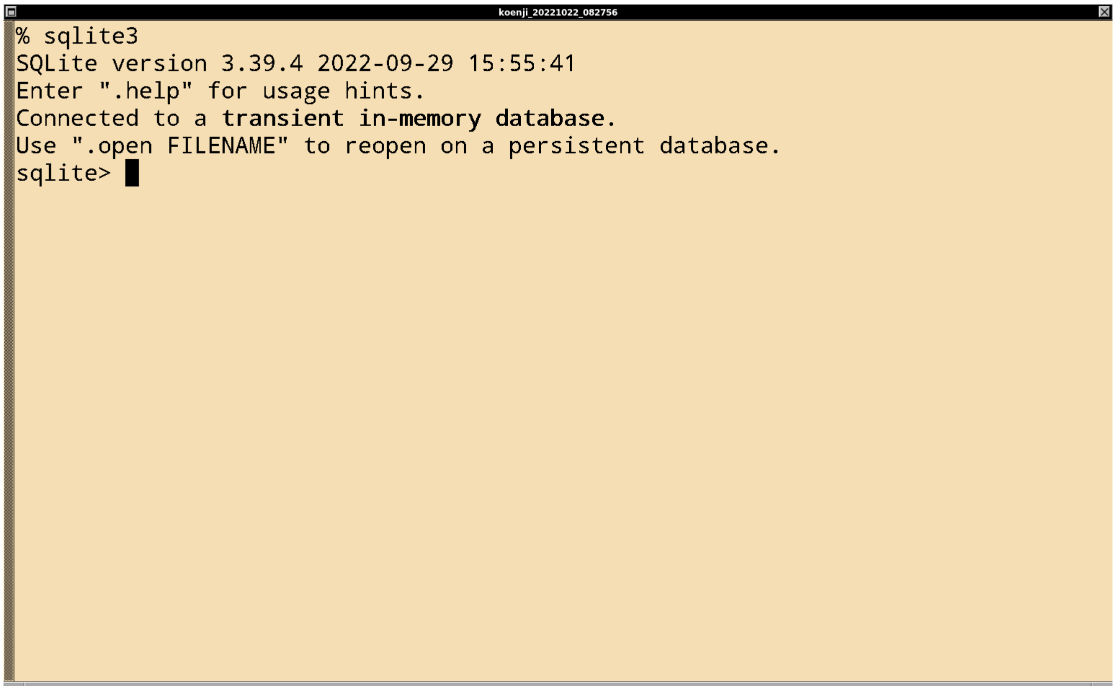


Figure 9: Starting SQLite command-line program on a terminal emulator.

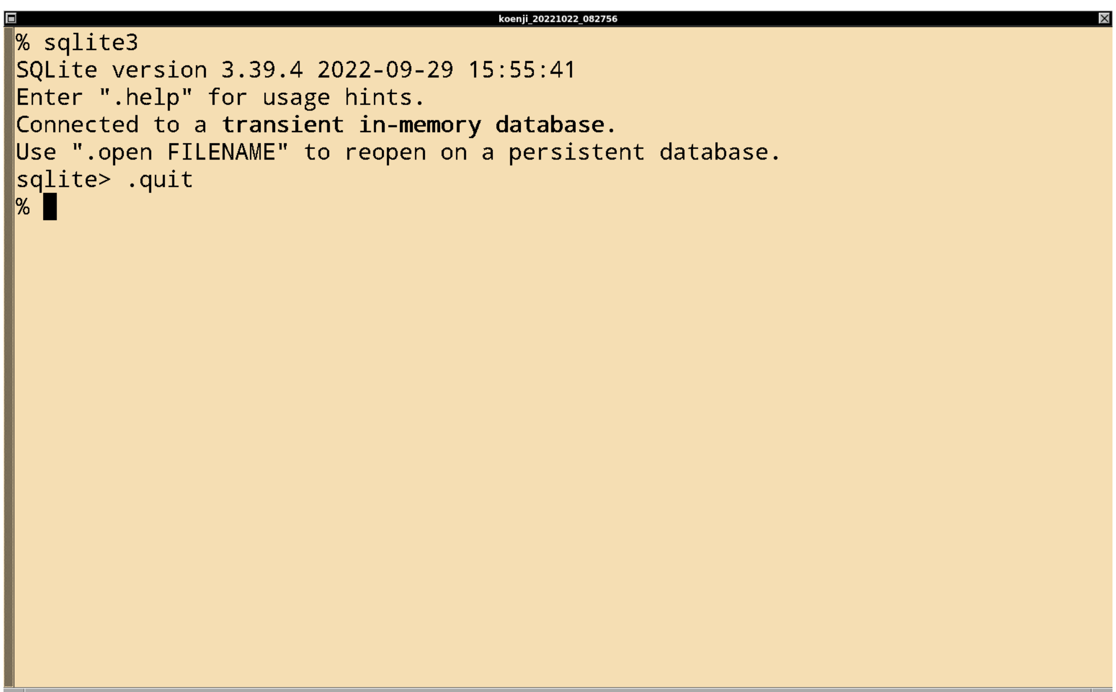


Figure 10: Quitting from SQLite command-line program on a terminal emulator.

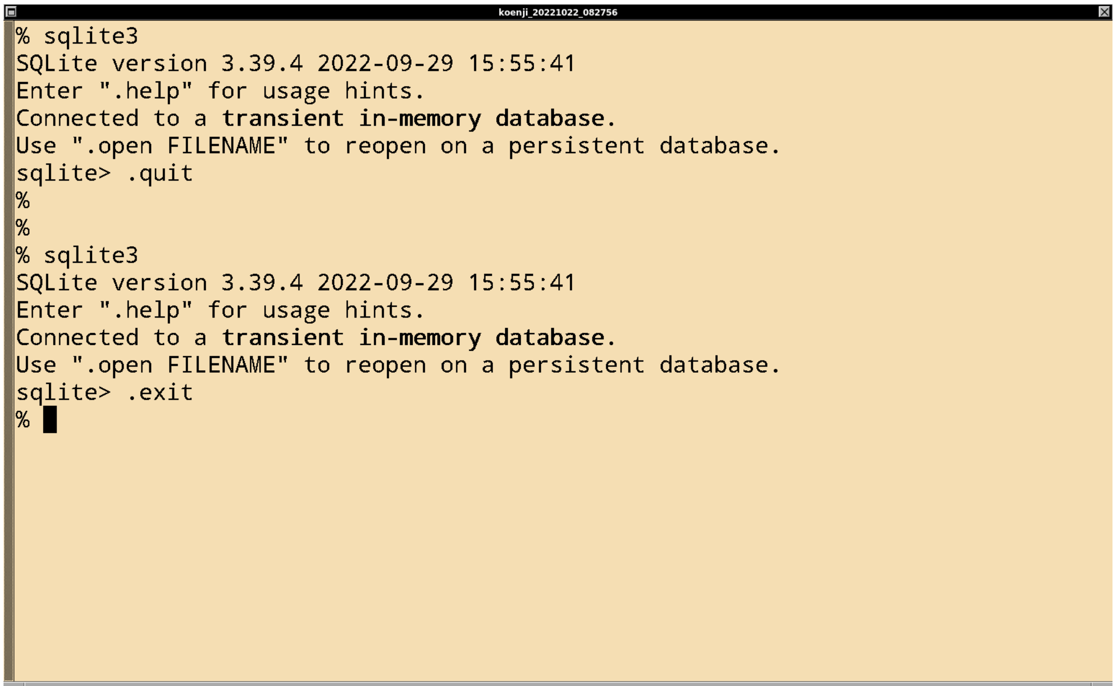


Figure 11: The other way to quit from SQLite command-line program on a terminal emulator.

2.3 Using SQLite on Binder

To use SQLite on Binder, start your favourite web browser. Then, visit following web page.

- `https://mybinder.org/v2/gh/kinoshitadaisuke/ncu_astroinformatics_202209/HEAD`

You see Jupyter Lab started on your web browser. (Fig. 12) You scroll down the page, and find an icon button for a terminal emulator named "Terminal". (Fig. 13) Give a double-click for the icon button "Terminal", then a terminal emulator starts. (Fig. 14) On a terminal emulator, type a command "`sqlite3`", then SQLite starts. (Fig. 15) When you quit from SQLite, type a command `.quit`. (Fig. 16)

2.4 Help command of SQLite

To learn about available commands of SQLite, try `.help` command. A list of available commands are shown.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .help
.archive ...             Manage SQL archives
.auth ON|OFF             Show authorizer callbacks
.backup ?DB? FILE        Backup DB (default "main") to FILE
.bail on|off             Stop after hitting an error.  Default OFF
.binary on|off           Turn binary output on or off.  Default OFF
.cd DIRECTORY            Change the working directory to DIRECTORY
.changes on|off          Show number of rows changed by SQL
.check GLOB              Fail if output since .testcase does not match
.clone NEWDB             Clone data into NEWDB from the existing database
.connection [close] [#]  Open or close an auxiliary database connection
.databases               List names and files of attached databases
.dbconfig ?op? ?val?     List or change sqlite3_db_config() options
.dbinfo ?DB?             Show status information about the database
```

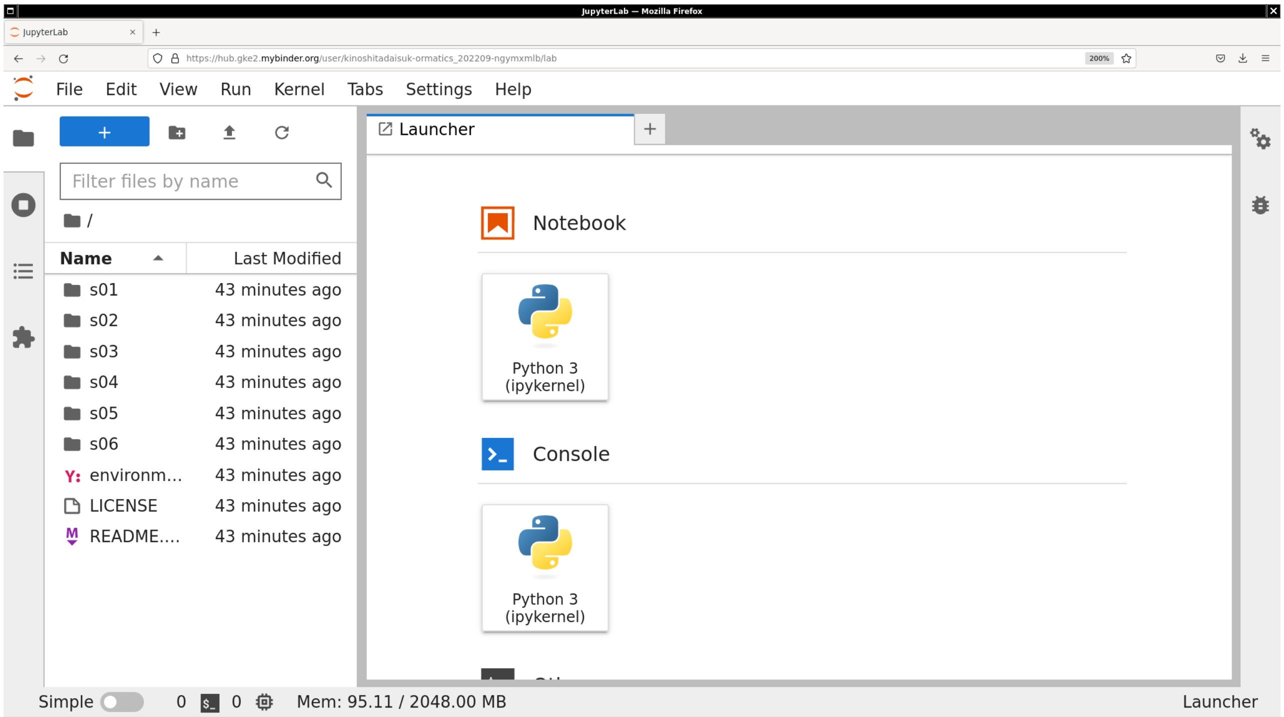


Figure 12: Jupyter Lab on Binder.

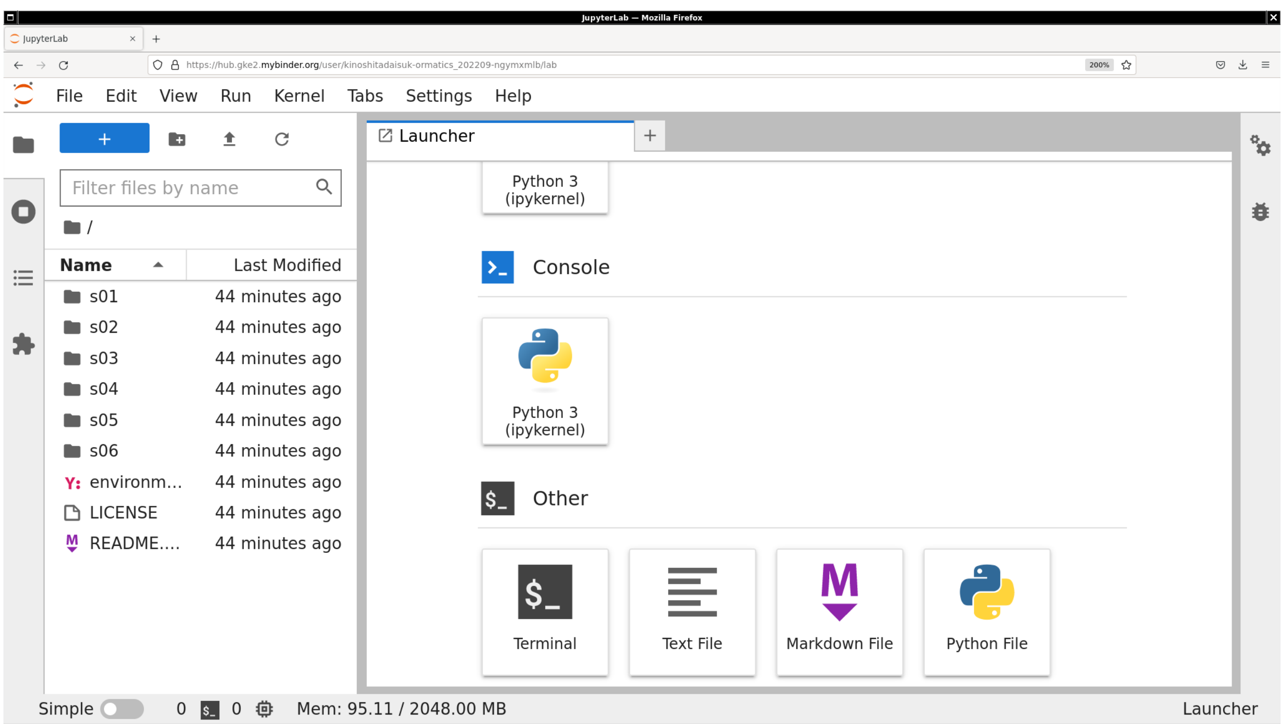


Figure 13: The icon button for starting a terminal emulator on Jupyter Lab on Binder.

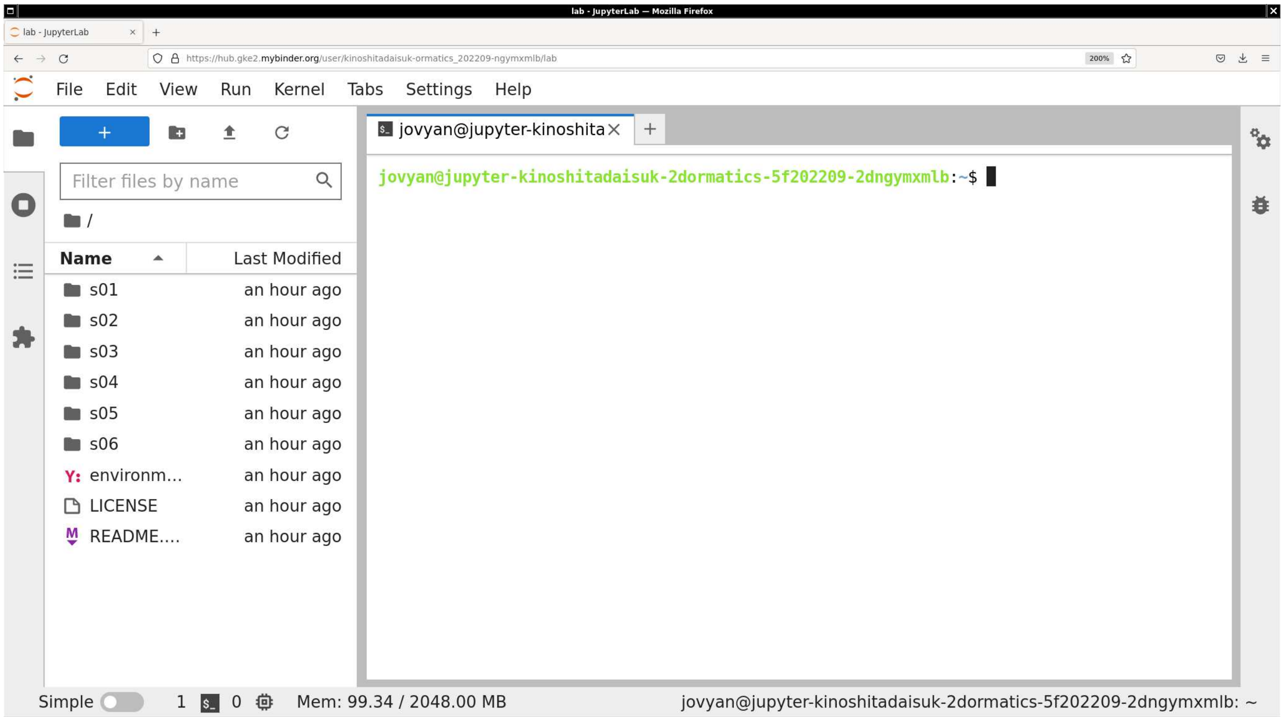


Figure 14: The terminal emulator on Jupyter Lab on Binder.

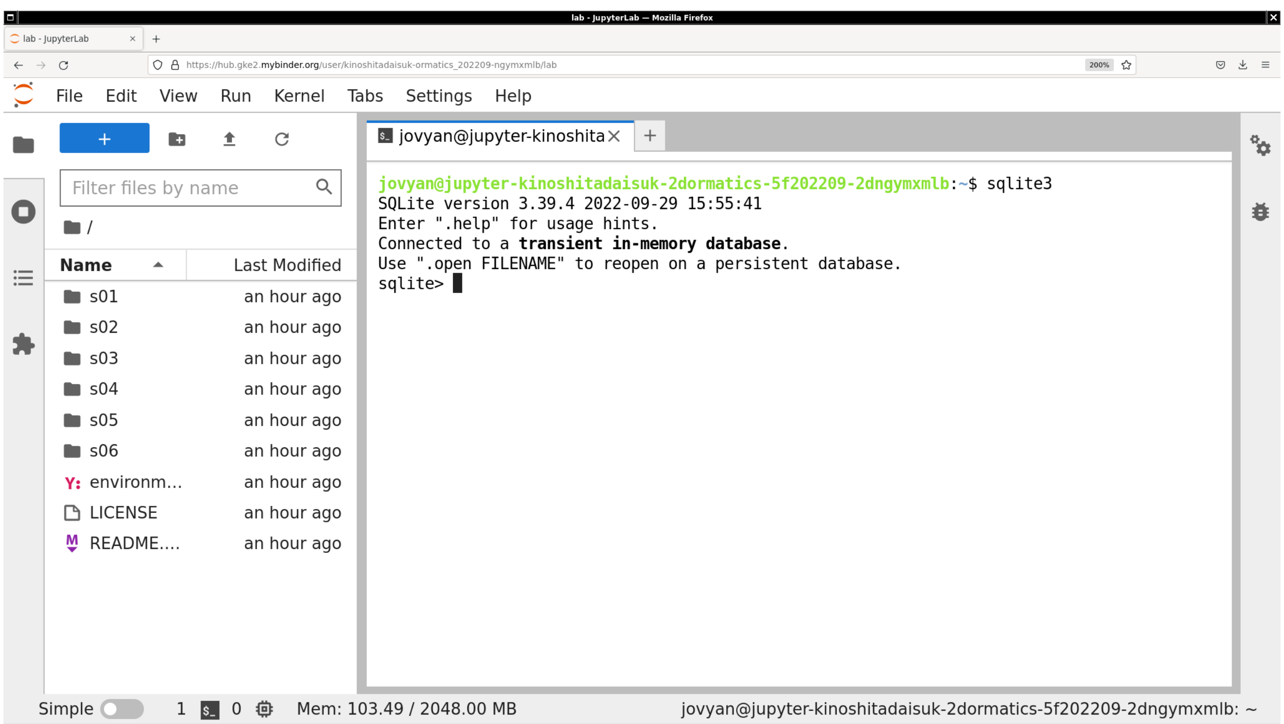


Figure 15: Starting SQLite on Jupyter Lab on Binder.

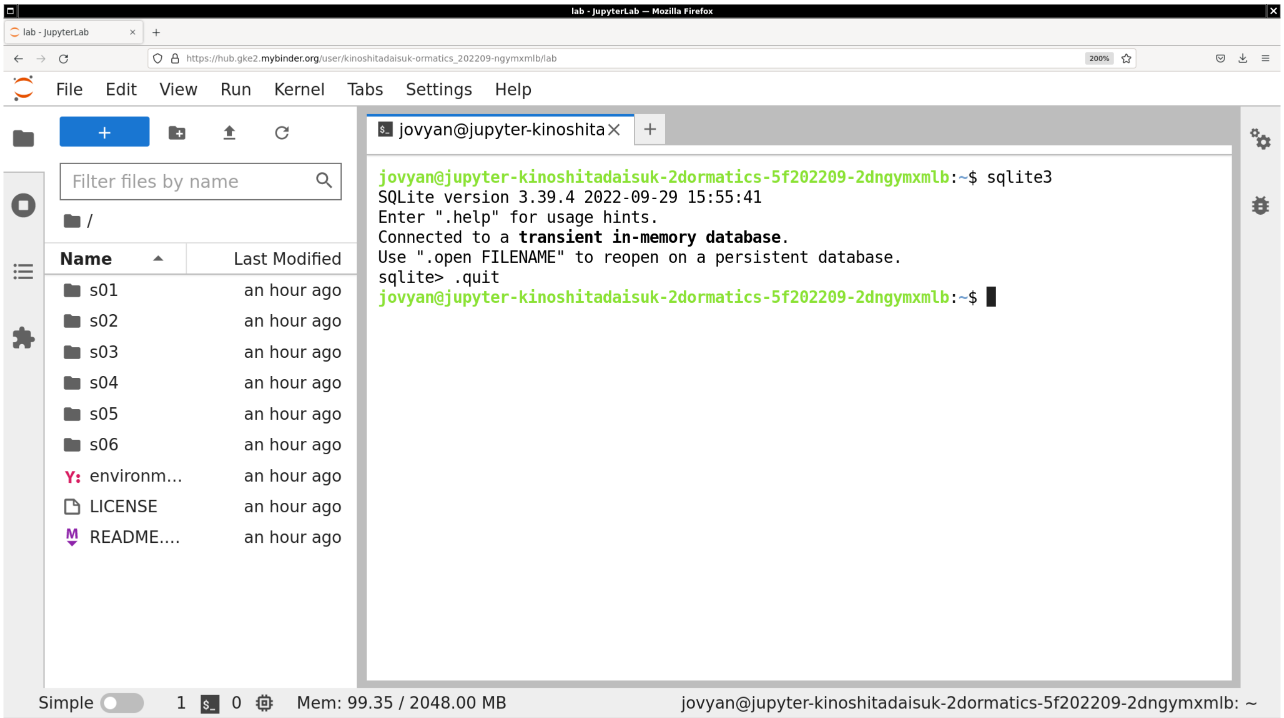


Figure 16: Quitting Jupyter Lab on Binder.

```
.dump ?OBJECTS?          Render database content as SQL
.echo on|off             Turn command echo on or off
.eqp on|off|full|...     Enable or disable automatic EXPLAIN QUERY PLAN
.excel                   Display the output of next command in spreadsheet
.exit ?CODE?             Exit this program with return-code CODE
.expert                  EXPERIMENTAL. Suggest indexes for queries
.explain ?on|off|auto?   Change the EXPLAIN formatting mode.  Default: auto
.filectrl CMD ...        Run various sqlite3_file_control() operations
.fullschema ?--indent?   Show schema and the content of sqlite_stat tables
.headers on|off          Turn display of headers on or off
.help ?-all? ?PATTERN?   Show help text for PATTERN
.import FILE TABLE       Import data from FILE into TABLE
.imposter INDEX TABLE    Create imposter table TABLE on index INDEX
.indexes ?TABLE?         Show names of indexes
.limit ?LIMIT? ?VAL?     Display or change the value of an SQLITE_LIMIT
.lint OPTIONS            Report potential schema issues.
.load FILE ?ENTRY?       Load an extension library
.log FILE|off            Turn logging on or off.  FILE can be stderr/stdout
.mode MODE ?OPTIONS?     Set output mode
.nonce STRING            Suspend safe mode for one command if nonce matches
.nullvalue STRING        Use STRING in place of NULL values
.once ?OPTIONS? ?FILE?   Output for the next SQL command only to FILE
.open ?OPTIONS? ?FILE?   Close existing database and reopen FILE
.output ?FILE?           Send output to FILE or stdout if FILE is omitted
.parameter CMD ...       Manage SQL parameter bindings
.print STRING...         Print literal STRING
.progress N              Invoke progress handler after every N opcodes
.prompt MAIN CONTINUE    Replace the standard prompts
.quit                    Exit this program
.read FILE               Read input from FILE or command output
.recover                 Recover as much data as possible from corrupt db.
.restore ?DB? FILE       Restore content of DB (default "main") from FILE
```

```
.save ?OPTIONS? FILE      Write database to FILE (an alias for .backup ...)
.scanstats on|off         Turn sqlite3_stmt_scanstatus() metrics on or off
.schema ?PATTERN?         Show the CREATE statements matching PATTERN
.selftest ?OPTIONS?       Run tests defined in the SELFTEST table
.separator COL ?ROW?      Change the column and row separators
.sha3sum ...              Compute a SHA3 hash of database content
.shell CMD ARGS...        Run CMD ARGS... in a system shell
.show                     Show the current values for various settings
.stats ?ARG?              Show stats or turn stats on or off
.system CMD ARGS...       Run CMD ARGS... in a system shell
.tables ?TABLE?           List names of tables matching LIKE pattern TABLE
.testcase NAME            Begin redirecting output to 'testcase-out.txt'
.testctrl CMD ...         Run various sqlite3_test_control() operations
.timeout MS               Try opening locked tables for MS milliseconds
.timer on|off             Turn SQL timer on or off
.trace ?OPTIONS?          Output each SQL statement as it is run
.vfsinfo ?AUX?            Information about the top-level VFS
.vfslist                  List all available VFSes
.vfsname ?AUX?            Print the name of the VFS stack
.width NUM1 NUM2 ...      Set minimum column widths for columnar output
```

If you would like to know more about the command `.show`, then try following.

```
% .help .show
.show                     Show the current values for various settings
```

Quit from SQLite.

```
sqlite> .quit
```

3 Making a small database

We now make a small database using the command-line program of SQLite.

3.1 Making a table

First, start SQLite.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Make a persistent database as a file on the hard disk (or SSD) on the computer. The command `.open` can be used to make an empty database. Use `.help` command to learn about `.open` command.

```
sqlite> .help .open
.open ?OPTIONS? ?FILE?    Close existing database and reopen FILE
     Options:
        --append         Use appendvfs to append database to the end of FILE
        --deserialize    Load into memory using sqlite3_deserialize()
        --hexdb          Load the output of "dbtotxt" as an in-memory db
```

```
        --maxsize N      Maximum size for --hexdb or --deserialized database
        --new            Initialize FILE to an empty database
        --nofollow       Do not follow symbolic links
        --readonly       Open FILE readonly
        --zip            FILE is a ZIP archive
```

Use `.open` command to make an empty database in a file "`solarsystem.db`".

```
sqlite> .open --new solarsystem.db
```

Use SQL command "create" to make a table for data of planets.

```
sqlite> create table planet (name text primary key, mass real, diameter real,
   ...> rotation_period real, orbital_period real, mean_temperature real,
   ...> satellite integer, ring text, magnetic_field text);
```

Use `.help` command to learn about the usage of `.tables` command.

```
sqlite> .help .tables
.tables ?TABLE?          List names of tables matching LIKE pattern TABLE
```

Try `.tables` command to show a list of existing tables.

```
sqlite> .tables
planet
```

The table "`planet`" does exist.
Use the command `.help` to learn about the usage of `.schema` command.

```
sqlite> .help .schema
.schema ?PATTERN?        Show the CREATE statements matching PATTERN
   Options:
      --indent              Try to pretty-print the schema
      --nosys               Omit objects whose names start with "sqlite_"
```

Use the command `.schema` to check the structure of the table "`planet`".

```
sqlite> .schema --indent planet
CREATE TABLE planet(
  name text primary key,
  mass real,
  diameter real,
  rotation_period real,
  orbital_period real,
  mean_temperature real,
  satellite integer,
  ring text,
  magnetic_field text
);
```

Add data of planets to the table "`planet`". Note that you need a semicolon (";") at the end of each command.

```
sqlite> insert into planet values ('Mercury', 3.30E23, 4.879E3,
   ...> 1407.6, 88.0, 167, 0, 'No', 'Yes');
sqlite> insert into planet values ('Venus', 4.87E24, 1.2104E4,
   ...> -5832.5, 224.7, 464, 0, 'No', 'No');
sqlite> insert into planet values ('Earth', 5.97E24, 1.2756E4,
   ...> 23.9, 365.2, 15, 1, 'No', 'Yes');
sqlite> insert into planet values ('Mars', 6.42E23, 6.792E3,
   ...> 24.6, 687.0, -65, 2, 'No', 'No');
sqlite> insert into planet values ('Jupiter', 1.898E27, 1.42984E5,
   ...> 9.9, 4331, -110, 79, 'Yes', 'Yes');
sqlite> insert into planet values ('Saturn', 5.68E26, 1.20536E5,
   ...> 10.7, 10747, -140, 82, 'Yes', 'Yes');
sqlite> insert into planet values ('Uranus', 8.68E25, 5.1118E4,
   ...> -17.2, 30589, -195, 27, 'Yes', 'Yes');
sqlite> insert into planet values ('Neptune', 1.02E26, 4.9528E4,
   ...> 16.1, 59800, -200, 14, 'Yes', 'Yes');
```

3.2 Trying SQL queries

Try following simple SQL query. Following example prints planet name, mass, diameter, number of satellites, existence/non-existence of ring system, existence/non-existence of global magnetic field for all the records in the table of the database.

```
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet;
Mercury|3.3e+23|4879.0|0|No|Yes
Venus|4.87e+24|12104.0|0|No|No
Earth|5.97e+24|12756.0|1|No|Yes
Mars|6.42e+23|6792.0|2|No|No
Jupiter|1.898e+27|142984.0|79|Yes|Yes
Saturn|5.68e+26|120536.0|82|Yes|Yes
Uranus|8.68e+25|51118.0|27|Yes|Yes
Neptune|1.02e+26|49528.0|14|Yes|Yes
```

This is not a user-friendly output format. Let us change some settings.
Use `.help` command to learn about the usage of `.show` command.

```
sqlite> .help .show
.show                    Show the current values for various settings
```

Try `.show` command to show current settings.

```
sqlite> .help .show
.show                    Show the current values for various settings
sqlite> .show
        echo: off
         eqp: off
     explain: auto
     headers: off
        mode: list
   nullvalue: ""
      output: stdout
colseparator: "|"
rowseparator: "\n"
       stats: off
       width:
    filename: solarsystem.db
```

Use `.help` command to learn about the usage of `.headers` command.

```
sqlite> .help headers
.headers on|off          Turn display of headers on or off
```

Turn on header using `.header` command.

```
sqlite> .headers on
```

Use `.show` command to check current settings.

```
sqlite> .show
        echo: off
         eqp: off
     explain: auto
     headers: on
        mode: list
   nullvalue: ""
      output: stdout
colseparator: "|"
rowseparator: "\n"
       stats: off
       width:
    filename: solarsystem.db
```

Use `.help` command to learn about the usage of `.mode` command.

```
sqlite> .help mode
.mode MODE ?OPTIONS?     Set output mode
   MODE is one of:
     ascii       Columns/rows delimited by 0x1F and 0x1E
     box         Tables using unicode box-drawing characters
     csv         Comma-separated values
     column      Output in columns.  (See .width)
     html        HTML <table> code
     insert      SQL insert statements for TABLE
     json        Results in a JSON array
     line        One value per line
     list        Values delimited by "|"
     markdown    Markdown table format
     qbox        Shorthand for "box --width 60 --quote"
     quote       Escape answers as for SQL
     table       ASCII-art table
     tabs        Tab-separated values
     tcl         TCL list elements
   OPTIONS: (for columnar modes or insert mode):
     --wrap N       Wrap output lines to no longer than N characters
     --wordwrap B   Wrap or not at word boundaries per B (on/off)
     --ww           Shorthand for "--wordwrap 1"
     --quote        Quote output text as SQL literals
     --noquote      Do not quote output text
     TABLE          The name of SQL table used for "insert" mode
```

Use `.mode` command to change the mode.

```
sqlite> .mode column
```

Use `.show` command to check current settings.

```
sqlite> .show
        echo: off
         eqp: off
     explain: auto
     headers: on
        mode: column --wrap 60 --wordwrap off --noquote
   nullvalue: ""
      output: stdout
colseparator: "|"
rowseparator: "\n"
       stats: off
       width:
    filename: solarsystem.db
```

Try SQL query again.

```
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet;
name     mass       diameter  satellite  ring  magnetic_field
-------  ---------  --------  ---------  ----  --------------
Mercury  3.3e+23    4879.0    0          No    Yes
Venus    4.87e+24   12104.0   0          No    No
Earth    5.97e+24   12756.0   1          No    Yes
Mars     6.42e+23   6792.0    2          No    No
Jupiter  1.898e+27  142984.0  79         Yes   Yes
Saturn   5.68e+26   120536.0  82         Yes   Yes
Uranus   8.68e+25   51118.0   27         Yes   Yes
Neptune  1.02e+26   49528.0   14         Yes   Yes
```

Now, output format looks much better.
Try the mode "`table`".

```
sqlite> .mode table
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet;
+---------+-----------+----------+-----------+------+----------------+
|  name   |   mass    | diameter | satellite | ring | magnetic_field |
+---------+-----------+----------+-----------+------+----------------+
| Mercury | 3.3e+23   | 4879.0   | 0         | No   | Yes            |
| Venus   | 4.87e+24  | 12104.0  | 0         | No   | No             |
| Earth   | 5.97e+24  | 12756.0  | 1         | No   | Yes            |
| Mars    | 6.42e+23  | 6792.0   | 2         | No   | No             |
| Jupiter | 1.898e+27 | 142984.0 | 79        | Yes  | Yes            |
| Saturn  | 5.68e+26  | 120536.0 | 82        | Yes  | Yes            |
| Uranus  | 8.68e+25  | 51118.0  | 27        | Yes  | Yes            |
| Neptune | 1.02e+26  | 49528.0  | 14        | Yes  | Yes            |
+---------+-----------+----------+-----------+------+----------------+
```

Try following practice.

Practice 06-01

Try a SQL query for the table "`planet`".

3.3 More about SQL queries

Sort the output by using "`order by`".

```
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet
   ...> order by diameter;
+---------+----------+----------+-----------+------+----------------+
|  name   |   mass   | diameter | satellite | ring | magnetic_field |
+---------+----------+----------+-----------+------+----------------+
| Mercury | 3.3e+23  | 4879.0   | 0         | No   | Yes            |
| Mars    | 6.42e+23 | 6792.0   | 2         | No   | No             |
| Venus   | 4.87e+24 | 12104.0  | 0         | No   | No             |
| Earth   | 5.97e+24 | 12756.0  | 1         | No   | Yes            |
| Neptune | 1.02e+26 | 49528.0  | 14        | Yes  | Yes            |
| Uranus  | 8.68e+25 | 51118.0  | 27        | Yes  | Yes            |
| Saturn  | 5.68e+26 | 120536.0 | 82        | Yes  | Yes            |
| Jupiter | 1.898e+27 | 142984.0 | 79       | Yes  | Yes            |
+---------+----------+----------+-----------+------+----------------+
```

Sort the output in descending order.

```
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet
   ...> order by diameter desc;
+---------+----------+----------+-----------+------+----------------+
|  name   |   mass   | diameter | satellite | ring | magnetic_field |
+---------+----------+----------+-----------+------+----------------+
| Jupiter | 1.898e+27 | 142984.0 | 79       | Yes  | Yes            |
| Saturn  | 5.68e+26 | 120536.0 | 82        | Yes  | Yes            |
| Uranus  | 8.68e+25 | 51118.0  | 27        | Yes  | Yes            |
| Neptune | 1.02e+26 | 49528.0  | 14        | Yes  | Yes            |
| Earth   | 5.97e+24 | 12756.0  | 1         | No   | Yes            |
| Venus   | 4.87e+24 | 12104.0  | 0         | No   | No             |
| Mars    | 6.42e+23 | 6792.0   | 2         | No   | No             |
| Mercury | 3.3e+23  | 4879.0   | 0         | No   | Yes            |
+---------+----------+----------+-----------+------+----------------+
```

Try following practice.

Practice 06-02

Try a SQL query for the table "`planet`" using "`order by`".

Search for planets having 10 or more satellites by using `where`.

```
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet
   ...> where satellite >= 10;
+---------+----------+----------+-----------+------+----------------+
|  name   |   mass   | diameter | satellite | ring | magnetic_field |
+---------+----------+----------+-----------+------+----------------+
| Jupiter | 1.898e+27 | 142984.0 | 79       | Yes  | Yes            |
| Saturn  | 5.68e+26 | 120536.0 | 82        | Yes  | Yes            |
| Uranus  | 8.68e+25 | 51118.0  | 27        | Yes  | Yes            |
| Neptune | 1.02e+26 | 49528.0  | 14        | Yes  | Yes            |
+---------+----------+----------+-----------+------+----------------+
```

Two ore more conditions can be used for `where`.

```
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet
   ...> where ( (mass >1e+26) and (diameter > 50000) );
+---------+----------+----------+-----------+------+----------------+
|  name   |   mass   | diameter | satellite | ring | magnetic_field |
+---------+----------+----------+-----------+------+----------------+
| Jupiter | 1.898e+27 | 142984.0 | 79        | Yes  | Yes            |
| Saturn  | 5.68e+26  | 120536.0 | 82        | Yes  | Yes            |
+---------+----------+----------+-----------+------+----------------+
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet
   ...> where ( (mass >1e+26) or (diameter > 50000) );
+---------+----------+----------+-----------+------+----------------+
|  name   |   mass   | diameter | satellite | ring | magnetic_field |
+---------+----------+----------+-----------+------+----------------+
| Jupiter | 1.898e+27 | 142984.0 | 79        | Yes  | Yes            |
| Saturn  | 5.68e+26  | 120536.0 | 82        | Yes  | Yes            |
| Uranus  | 8.68e+25  | 51118.0  | 27        | Yes  | Yes            |
| Neptune | 1.02e+26  | 49528.0  | 14        | Yes  | Yes            |
+---------+----------+----------+-----------+------+----------------+
```

Find planets with ring system.

```
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet
   ...> where ring is 'Yes';
+---------+----------+----------+-----------+------+----------------+
|  name   |   mass   | diameter | satellite | ring | magnetic_field |
+---------+----------+----------+-----------+------+----------------+
| Jupiter | 1.898e+27 | 142984.0 | 79        | Yes  | Yes            |
| Saturn  | 5.68e+26  | 120536.0 | 82        | Yes  | Yes            |
| Uranus  | 8.68e+25  | 51118.0  | 27        | Yes  | Yes            |
| Neptune | 1.02e+26  | 49528.0  | 14        | Yes  | Yes            |
+---------+----------+----------+-----------+------+----------------+
sqlite> select name,mass,diameter,satellite,ring,magnetic_field from planet
   ...> where ring like 'Y%';
+---------+----------+----------+-----------+------+----------------+
|  name   |   mass   | diameter | satellite | ring | magnetic_field |
+---------+----------+----------+-----------+------+----------------+
| Jupiter | 1.898e+27 | 142984.0 | 79        | Yes  | Yes            |
| Saturn  | 5.68e+26  | 120536.0 | 82        | Yes  | Yes            |
| Uranus  | 8.68e+25  | 51118.0  | 27        | Yes  | Yes            |
| Neptune | 1.02e+26  | 49528.0  | 14        | Yes  | Yes            |
+---------+----------+----------+-----------+------+----------------+
```

Try following practice.

Practice 06-03

Try a SQL query for the table "`planet`" using "`where`".

Quit from SQLite.

```
sqlite> .quit
```

Now, you have a file named "`solarsystem.db`".

```
% ls -lF
total 1
-rw-r--r--  1 daisuke  taiwan   2291 Oct 20 07:30 ai202209_s06.ipynb
-rw-r--r--  1 daisuke  taiwan  12288 Oct 21 08:49 solarsystem.db
% file solarsystem.db
solarsystem.db: SQLite 3.x database, last written using SQLite version 3039004,
file counter 9, database pages 3, cookie 0x1, schema 4, UTF-8, version-valid-for
 9
```

3.4 Opening an existing database file

Start SQLite.

```
% ls
ai202209_s06.ipynb  solarsystem.db
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite>
```

Use `.open` command to open an existing database file.

```
sqlite> .open solarsystem.db
```

Now, the database file "`solarsystem.db`" is opened. Check available table.

```
sqlite> .tables
planet
```

The table "`planet`" is available. Check the structure of the table "`planet`".

```
sqlite> .schema --indent planet
CREATE TABLE planet(
  name text primary key,
  mass real,
  diameter real,
  rotation_period real,
  orbital_period real,
  mean_temperature real,
  satellite integer,
  ring text,
  magnetic_field text
);
```

Try SQL queries.

```
sqlite> .headers on
sqlite> .mode table
sqlite> select name, mass, diameter, mean_temperature from planet
   ...> order by mass desc;
+---------+----------+---------+-----------------+
```

```
|  name   |   mass    | diameter | mean_temperature |
+---------+-----------+----------+------------------+
| Jupiter | 1.898e+27 | 142984.0 | -110.0           |
| Saturn  | 5.68e+26  | 120536.0 | -140.0           |
| Neptune | 1.02e+26  | 49528.0  | -200.0           |
| Uranus  | 8.68e+25  | 51118.0  | -195.0           |
| Earth   | 5.97e+24  | 12756.0  | 15.0             |
| Venus   | 4.87e+24  | 12104.0  | 464.0            |
| Mercury | 3.3e+23   | 4879.0   | 167.0            |
| Mars    | 6.42e+23  | 6792.0   | -65.0            |
+---------+-----------+----------+------------------+
sqlite> select name, mass, diameter, rotation_period, orbital_period
   ...> from planet where orbital_period < 1000 order by orbital_period desc;
+---------+----------+----------+-----------------+----------------+
|  name   |   mass   | diameter | rotation_period | orbital_period |
+---------+----------+----------+-----------------+----------------+
| Mars    | 6.42e+23 | 6792.0   | 24.6            | 687.0          |
| Earth   | 5.97e+24 | 12756.0  | 23.9            | 365.2          |
| Venus   | 4.87e+24 | 12104.0  | -5832.5         | 224.7          |
| Mercury | 3.3e+23  | 4879.0   | 1407.6          | 88.0           |
+---------+----------+----------+-----------------+----------------+
sqlite> select name, mass, diameter, mean_temperature, magnetic_field
   ...> from planet where ( mean_temperature > 0 and magnetic_field = 'Yes');
+---------+----------+----------+------------------+----------------+
|  name   |   mass   | diameter | mean_temperature | magnetic_field |
+---------+----------+----------+------------------+----------------+
| Mercury | 3.3e+23  | 4879.0   | 167.0            | Yes            |
| Earth   | 5.97e+24 | 12756.0  | 15.0             | Yes            |
+---------+----------+----------+------------------+----------------+
```

Quit from SQLite.

```
sqlite> .quit
```

Here is the other way to open an existing database file.

```
% sqlite3 solarsystem.db
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
sqlite> .tables
planet
sqlite> .schema --indent planet
CREATE TABLE planet(
  name text primary key,
  mass real,
  diameter real,
  rotation_period real,
  orbital_period real,
  mean_temperature real,
  satellite integer,
  ring text,
  magnetic_field text
);
```

Try a SQL query.

```
sqlite> .headers on
sqlite> .mode table
sqlite> select name, mass, diameter, mean_temperature, satellite from planet
   ...> where satellite > 0 order by mean_temperature desc;
+---------+----------+----------+------------------+-----------+
|  name   |   mass   | diameter | mean_temperature | satellite |
+---------+----------+----------+------------------+-----------+
| Earth   | 5.97e+24 | 12756.0  | 15.0             | 1         |
| Mars    | 6.42e+23 | 6792.0   | -65.0            | 2         |
| Jupiter | 1.898e+27 | 142984.0 | -110.0          | 79        |
| Saturn  | 5.68e+26 | 120536.0 | -140.0           | 82        |
| Uranus  | 8.68e+25 | 51118.0  | -195.0           | 27        |
| Neptune | 1.02e+26 | 49528.0  | -200.0           | 14        |
+---------+----------+----------+------------------+-----------+
```

3.5 Importing data from a CSV file

Download a CSV (Comma Separated Values) file. Here is a Python script to download CSV file "`dp.csv`".

Python Code 1: ai202209_s06_00.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/21 14:18:19 (CST) daisuke>
#

# importing urllib module
import urllib.request

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# URL of data file
url_data = 'https://s3b.astro.ncu.edu.tw/ai_202209/data/dp.csv'

# output file name
file_output = 'dwarf_planet.csv'

# printing status
print (f'Now, fetching the file {url_data}...')

# opening URL
with urllib.request.urlopen (url_data) as fh_read:
    # reading data
    data_byte = fh_read.read ()

# printing status
print (f'Finished fetching the file {url_data}!')

# converting raw byte data into string
data_str = data_byte.decode ('utf-8')

# printing status
print (f'Now, writing the data into file {file_output}...')
```

```
# opening file for writing
with open (file_output, 'w') as fh_write:
    # writing data
    fh_write.write (data_str)

# printing status
print (f'Finished writing the data into file {file_output}!')
```

Execute above script to download CSV file.

```
% chmod a+x ai202209_s06_00.py
% ./ai202209_s06_00.py
Now, fetching the file https://s3b.astro.ncu.edu.tw/ai_202209/data/dp.csv...
Finished fetching the file https://s3b.astro.ncu.edu.tw/ai_202209/data/dp.csv!
Now, writing the data into file dp.csv...
Finished writing the data into file dp.csv!
% ls -lF dwarf_planet.csv
-rw-r--r--  1 daisuke  taiwan  452 Oct 21 14:20 dwarf_planet.csv
% file dwarf_planet.csv
dwarf_planet.csv: ASCII text
% cat dwarf_planet.csv
# dwarf planet database
#
# data format:
#   name, a, e, i, q, Q, P, H
#
# Ref.: https://minorplanetcenter.net/dwarf_planets
#
(1) Ceres        ,  2.77, 0.08, 10.6,  2.55,  2.98,   4.60,  3.3
(134340) Pluto   , 39.67, 0.25, 17.1, 29.80, 49.54, 250   , -0.4
(136199) Eris    , 68.12, 0.43, 43.8, 38.69, 97.54, 562   , -1.2
(136472) Makemake, 45.26, 0.17, 29.0, 37.74, 52.78, 304   , -0.2
(136108) Haumea  , 42.94, 0.20, 28.2, 34.36, 51.52, 281   ,  0.2
```

Start SQLite and make a table.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> create table dwarfplanet (name text primary key, a real, e real,
   ...> i real, perihelion real, aphelion real, P real, H real);
sqlite> .tables
dwarfplanet
sqlite> .schema --indent dwarfplanet
CREATE TABLE dwarfplanet(
  name text primary key,
  a real,
  e real,
  i real,
  perihelion real,
  aphelion real,
  P real,
  H real
);
```

Use the command `.help` to learn about the usage of `.import` command.

```
sqlite> .help import
.import FILE TABLE       Import data from FILE into TABLE
   Options:
     --ascii               Use \037 and \036 as column and row separators
     --csv                 Use , and \n as column and row separators
     --skip N              Skip the first N rows of input
     --schema S            Target table to be S.TABLE
     -v                    "Verbose" - increase auxiliary output
   Notes:
     *  If TABLE does not exist, it is created.  The first row of input
        determines the column names.
     *  If neither --csv or --ascii are used, the input mode is derived
        from the ".mode" output mode
     *  If FILE begins with "|" then it is a command that generates the
        input text.
```

Use the command `.import` to import data from a CSV file. There are 7 lines of header in the CSV file. Add an option "`--skip 7`" to the `.import` command.

```
sqlite> .mode csv
sqlite> .separator ,
sqlite> .import --skip 7 dwarf_planet.csv dwarfplanet
```

The same thing can be done by following.

```
sqlite> create table dwarfplanet2 (name text primary key, a real, e real,
   ...> i real, perihelion real, aphelion real, P real, H real);
sqlite> .tables
dwarfplanet   dwarfplanet2
sqlite> .schema --indent dwarfplanet2
CREATE TABLE dwarfplanet2(
  name text primary key,
  a real,
  e real,
  i real,
  perihelion real,
  aphelion real,
  P real,
  H real
);
sqlite> .import --csv --skip 7 dwarf_planet.csv dwarfplanet2
```

The same thing can also be done by following.

```
sqlite> create table dwarfplanet3 (name text primary key, a real, e real,
   ...> i real, perihelion real, aphelion real, P real, H real);
sqlite> .tables
dwarfplanet   dwarfplanet2  dwarfplanet3
sqlite> .schema --indent dwarfplanet3
CREATE TABLE dwarfplanet3(
  name text primary key,
  a real,
  e real,
```

```
  i real,
  perihelion real,
  aphelion real,
  P real,
  H real
);
sqlite> .import --csv '| grep -v \# dwarf_planet.csv' dwarfplanet3
```

Save the database into a file.

```
sqlite> .save dwarf_planet.db
```

Try SQL queries.

```
sqlite> select name, a, e, i, H from dwarfplanet order by H;
+-------------------+-------+------+------+------+
|       name        |   a   |  e   |  i   |  H   |
+-------------------+-------+------+------+------+
| (136199) Eris     | 68.12 | 0.43 | 43.8 | -1.2 |
| (134340) Pluto    | 39.67 | 0.25 | 17.1 | -0.4 |
| (136472) Makemake | 45.26 | 0.17 | 29.0 | -0.2 |
| (136108) Haumea   | 42.94 | 0.2  | 28.2 | 0.2  |
| (1) Ceres         | 2.77  | 0.08 | 10.6 | 3.3  |
+-------------------+-------+------+------+------+
sqlite> select name, a, e, i, perihelion, aphelion from dwarfplanet
   ...> order by aphelion desc;
+-------------------+-------+------+------+------------+----------+
|       name        |   a   |  e   |  i   | perihelion | aphelion |
+-------------------+-------+------+------+------------+----------+
| (136199) Eris     | 68.12 | 0.43 | 43.8 | 38.69      | 97.54    |
| (136472) Makemake | 45.26 | 0.17 | 29.0 | 37.74      | 52.78    |
| (136108) Haumea   | 42.94 | 0.2  | 28.2 | 34.36      | 51.52    |
| (134340) Pluto    | 39.67 | 0.25 | 17.1 | 29.8       | 49.54    |
| (1) Ceres         | 2.77  | 0.08 | 10.6 | 2.55       | 2.98     |
+-------------------+-------+------+------+------------+----------+
```

Try following practice.

Practice 06-04

Try a SQL query for the table "`dwarfplanet`".

Quit from SQLite.

```
sqlite> .quit
```

Now, you have a file named "`dwarf_planet.db`".

```
% ls -lF *.db
-rw-r--r--  1 daisuke  taiwan  28672 Oct 21 14:57 dwarf_planet.db
-rw-r--r--  1 daisuke  taiwan  12288 Oct 21 08:49 solarsystem.db
```

3.6 Exporting database into a SQL file

Open a database file.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open dwarf_planet.db
sqlite> .tables
dwarfplanet   dwarfplanet2  dwarfplanet3
```

Use the command `.output` to set output file name.

```
sqlite> .output dwarf_planet.sql
```

Use the command `.dump` to export the database into a file.

```
sqlite> .dump
```

Quit from SQLite.

```
sqlite> .quit
```

Now, you have a file named "`dwarf_planet.sql`".

```
% ls -lF *.sql
-rw-r--r--  1 daisuke  taiwan  3253 Oct 21 15:06 dwarf_planet.sql
% file dwarf_planet.sql
dwarf_planet.sql: ASCII text
% head dwarf_planet.sql | cut -b 1-80
PRAGMA foreign_keys=OFF;
BEGIN TRANSACTION;
CREATE TABLE dwarfplanet (name text primary key, a real, e real,
i real, perihelion real, aphelion real, P real, H real);
INSERT INTO dwarfplanet VALUES('(1) Ceres        ',2.7700000000000000177,0.08000
INSERT INTO dwarfplanet VALUES('(134340) Pluto   ',39.670000000000001706,0.25,17
INSERT INTO dwarfplanet VALUES('(136199) Eris    ',68.120000000000004549,0.42999
INSERT INTO dwarfplanet VALUES('(136472) Makemake',45.259999999999999801,0.170000
INSERT INTO dwarfplanet VALUES('(136108) Haumea  ',42.939999999999997727,0.20000
CREATE TABLE dwarfplanet2 (name text primary key, a real, e real,
```

3.7 Reading a SQL file

Start SQLite, and read a SQL file.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .tables
sqlite> .read dwarf_planet.sql
sqlite> .tables
dwarfplanet   dwarfplanet2  dwarfplanet3
```

Try a SQL query.

```
sqlite> .headers on
sqlite> .mode table
sqlite> select name, a, e, i, P, H from dwarfplanet3 order by P;
+-------------------+-------+------+------+-------+------+
|       name        |   a   |  e   |  i   |   P   |  H   |
+-------------------+-------+------+------+-------+------+
| (1) Ceres         | 2.77  | 0.08 | 10.6 | 4.6   | 3.3  |
| (134340) Pluto    | 39.67 | 0.25 | 17.1 | 250.0 | -0.4 |
| (136108) Haumea   | 42.94 | 0.2  | 28.2 | 281.0 | 0.2  |
| (136472) Makemake | 45.26 | 0.17 | 29.0 | 304.0 | -0.2 |
| (136199) Eris     | 68.12 | 0.43 | 43.8 | 562.0 | -1.2 |
+-------------------+-------+------+------+-------+------+
```

Quit from SQLite.

```
sqlite> .quit
```

3.8 Exporting database into a CSV file

Start SQLite and read SQL file.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .read dwarf_planet.sql
sqlite> .tables
dwarfplanet   dwarfplanet2  dwarfplanet3
```

Use the command `.help` to learn about the usage of `.once` command.

```
sqlite> .help .once
.once ?OPTIONS? ?FILE?   Output for the next SQL command only to FILE
     If FILE begins with '|' then open as a pipe
       --bom  Put a UTF8 byte-order mark at the beginning
       -e     Send output to the system text editor
       -x     Send output as CSV to a spreadsheet (same as ".excel")
```

Use the command `.once` to export the database into a CSV file.

```
sqlite> .headers on
sqlite> .mode csv
sqlite> .once new.csv
sqlite> select * from dwarfplanet;
```

Quit from SQLite.

```
sqlite> .quit
```

Now, you have a file named “`new.csv`”.

```
% ls -lF new.csv
-rw-r--r--  1 daisuke  taiwan  327 Oct 21 15:19 new.csv
% file new.csv
new.csv: CSV text
% cat new.csv
name,a,e,i,perihelion,aphelion,P,H
"(1) Ceres        ",2.77,0.08,10.6,2.55,2.98,4.6,3.3
"(134340) Pluto   ",39.67,0.25,17.1,29.8,49.54,250.0,-0.4
"(136199) Eris    ",68.12,0.43,43.8,38.69,97.54,562.0,-1.2
"(136472) Makemake",45.26,0.17,29.0,37.74,52.78,304.0,-0.2
"(136108) Haumea  ",42.94,0.2,28.2,34.36,51.52,281.0,0.2
```

Use your favourite spreadsheet program to visualise the CSV file. Here is an example of using the program "gnumeric" for viewing CSV file. (Fig. 17)

```
% gnumeric new.csv
```

About the program "gnumeric", visit following website to learn about it. (Fig. 18)

- gnumeric: `http://www.gnumeric.org/`

Or, you may use the program like LibreOffice. (Fig. 19)

- LibreOffice: `https://www.libreoffice.org/`

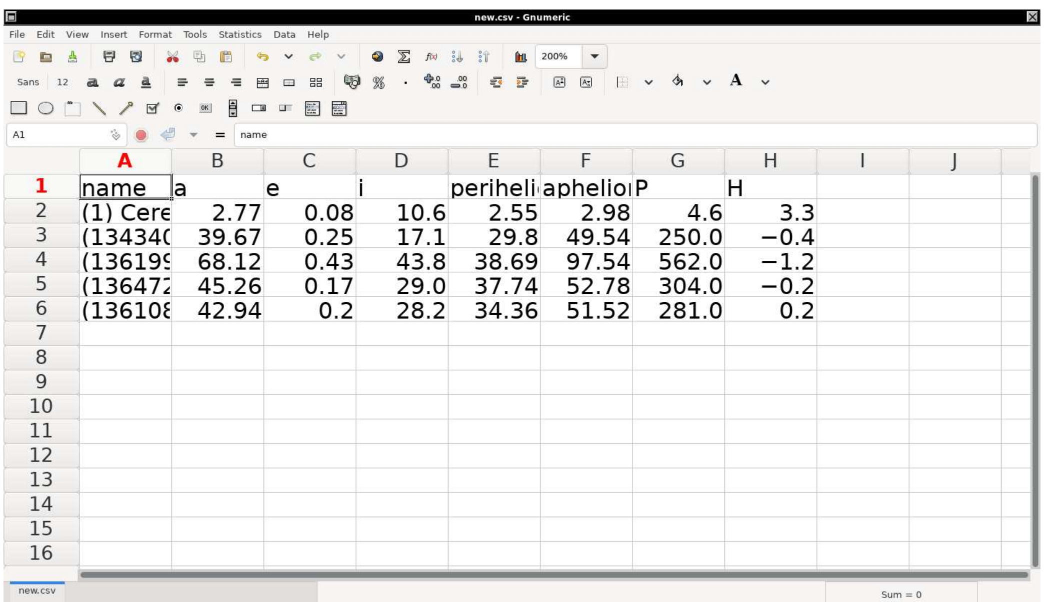


Figure 17: The CSV file "new.csv" opened by the program "gnumeric".

4 Constructing element database

We download the data of periodic table, and construct the element database.

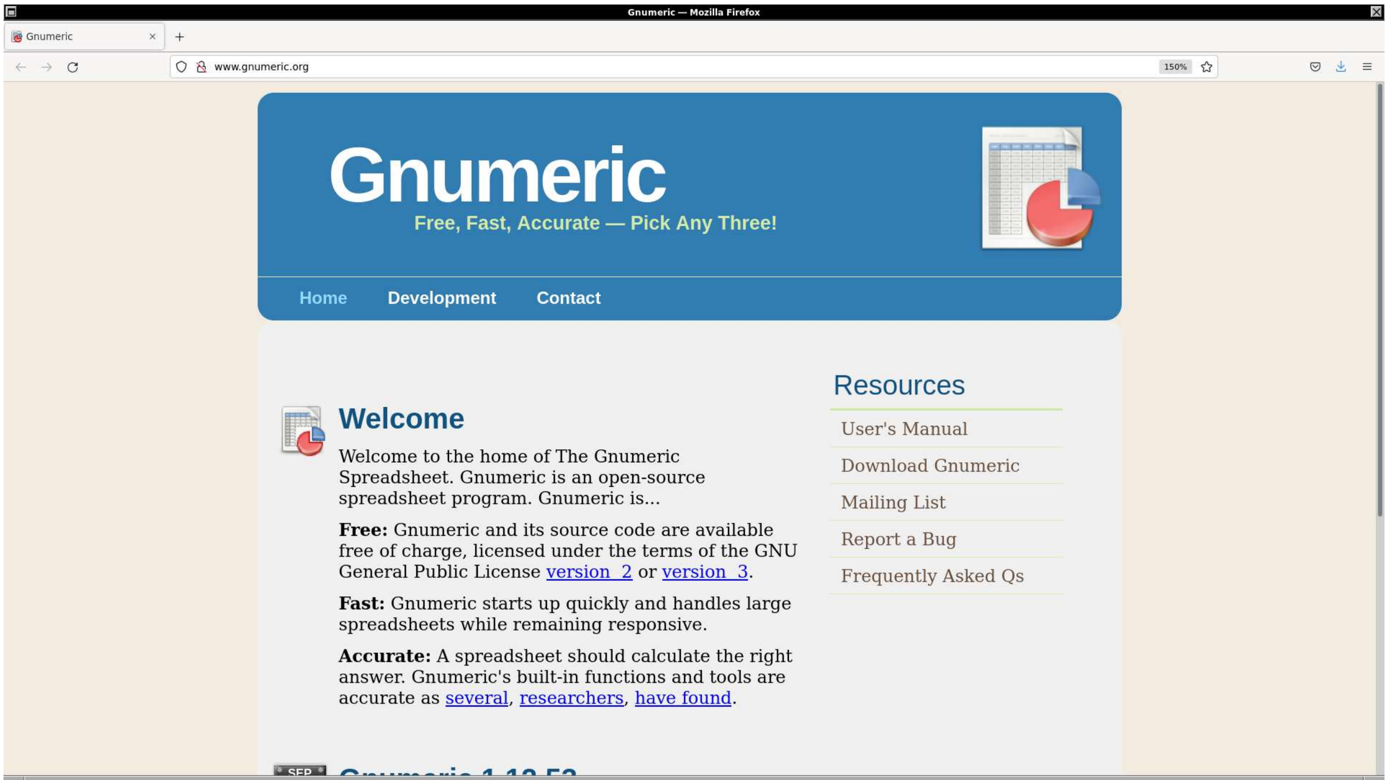


Figure 18: The official website of the program "`gnumeric`".

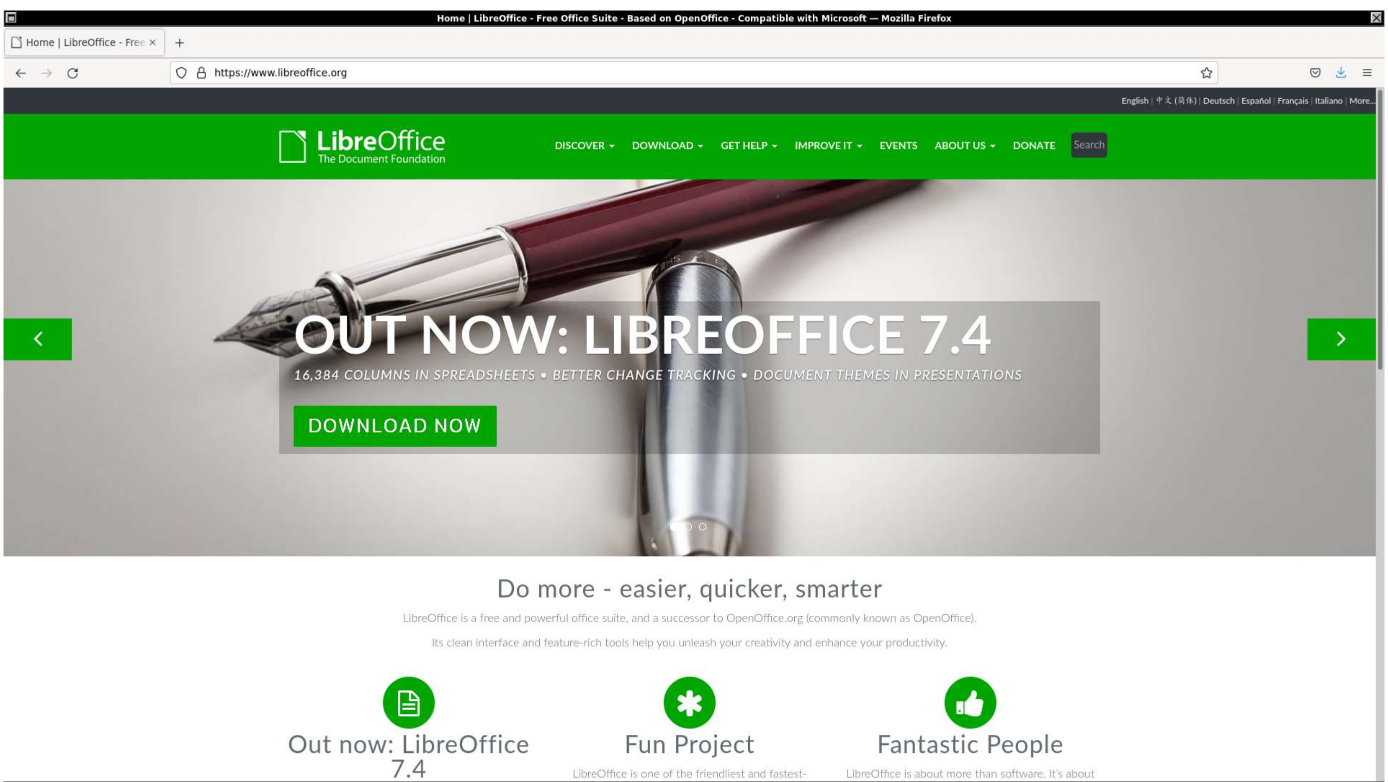


Figure 19: The official website of the program LibreOffice.

4.1 Downloading CSV file

Download the CSV file for the periodic table. Here is a Python script for downloading.

Python Code 2: ai202209_s06_01.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/21 15:44:39 (CST) daisuke>
#

# importing urllib module
import urllib.request

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# URL of data file
url_data = 'https://pubchem.ncbi.nlm.nih.gov/rest/pug/periodictable/CSV/'

# output file name
file_output = 'periodictable.csv'

# printing status
print (f'Now, fetching {url_data}...')

# opening URL
with urllib.request.urlopen (url_data) as fh_read:
    # reading data
    data_byte = fh_read.read ()

# printing status
print (f'Finished fetching {url_data}!')

# converting raw byte data into string
data_str = data_byte.decode ('utf-8')

# printing status
print (f'Now, writing data into file "{file_output}"...')

# opening file for writing
with open (file_output, 'w') as fh_write:
    # writing data
    fh_write.write (data_str)

# printing status
print (f'Finished writing data into file "{file_output}"!')
```

Execute above script to download CSV file.

```
% chmod a+x ai202209_s06_01.py
% ./ai202209_s06_01.py
Now, fetching https://pubchem.ncbi.nlm.nih.gov/rest/pug/periodictable/CSV/...
Finished fetching https://pubchem.ncbi.nlm.nih.gov/rest/pug/periodictable/CSV/!
Now, writing data into file "periodictable.csv"...
Finished writing data into file "periodictable.csv"!
% ls -lF periodictable.csv
```

```
-rw-r--r--  1 daisuke  taiwan  15016 Oct 21 15:44 periodictable.csv
% file periodictable.csv
periodictable.csv: CSV text
% head periodictable.csv | cut -b 1-76
"AtomicNumber","Symbol","Name","AtomicMass","CPKHexColor","ElectronConfigura
1,"H","Hydrogen",1.0080,"FFFFFF","1s1",2.2,120,13.598,0.754,"+1, -1","Gas",1
2,"He","Helium",4.00260,"D9FFFF","1s2","",140,24.587,"","0","Gas",0.95,4.22,
3,"Li","Lithium",7.0,"CC80FF","[He]2s1",0.98,182,5.392,0.618,"+1","Solid",45
4,"Be","Beryllium",9.012183,"C2FF00","[He]2s2",1.57,153,9.323,"","+2","Solid
5,"B","Boron",10.81,"FFB5B5","[He]2s2 2p1",2.04,192,8.298,0.277,"+3","Solid"
6,"C","Carbon",12.011,"909090","[He]2s2 2p2",2.55,170,11.260,1.263,"+4, +2,
7,"N","Nitrogen",14.007,"3050F8","[He] 2s2 2p3",3.04,155,14.534,"","+5, +4,
8,"O","Oxygen",15.999,"FF0D0D","[He]2s2 2p4",3.44,152,13.618,1.461,"-2","Gas
9,"F","Fluorine",18.99840316,"90E050","[He]2s2 2p5",3.98,135,17.423,3.339,"-
```

4.2 Importing CSV file and constructing database

Start SQLite and create a table.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> create table element (AtomicNumber integer primary key, Symbol text,
   ...> Name text, AtomicMass real, CPKHexColor text,
   ...> ElectronConfiguration text, Electronegativity real, AtomicRadius real,
   ...> IonizationEnergy real, ElectronAffinity real, OxidationStates text,
   ...> StandardState text, MeltingPoint real, BoilingPoint real,
   ...> Density real, GroupBlock text, YearDiscovered text);
sqlite> .tables
element
sqlite> .schema --indent element
CREATE TABLE element(
  AtomicNumber integer primary key,
  Symbol text,
  Name text,
  AtomicMass real,
  CPKHexColor text,
  ElectronConfiguration text,
  Electronegativity real,
  AtomicRadius real,
  IonizationEnergy real,
  ElectronAffinity real,
  OxidationStates text,
  StandardState text,
  MeltingPoint real,
  BoilingPoint real,
  Density real,
  GroupBlock text,
  YearDiscovered text
);
```

Import CSV file.

```
sqlite> .import --csv --skip 1 periodictable.csv element
```

Try SQL queries.

```
sqlite> .headers on
sqlite> .mode table
sqlite> select Symbol, Name, AtomicMass, StandardState, MeltingPoint,
   ...> BoilingPoint from element where StandardState is 'Liquid';
+--------+---------+------------+---------------+--------------+--------------+
| Symbol |  Name   | AtomicMass | StandardState | MeltingPoint | BoilingPoint |
+--------+---------+------------+---------------+--------------+--------------+
| Br     | Bromine | 79.9       | Liquid        | 265.95       | 331.95       |
| Hg     | Mercury | 200.59     | Liquid        | 234.32       | 629.88       |
+--------+---------+------------+---------------+--------------+--------------+
sqlite> select AtomicNumber, Name, Symbol, StandardState, Density from element
   ...> where Density >= 15.0 and Density != '' order by Density desc;
+--------------+--------------+--------+---------------+---------+
| AtomicNumber |     Name     | Symbol | StandardState | Density |
+--------------+--------------+--------+---------------+---------+
| 76           | Osmium       | Os     | Solid         | 22.57   |
| 77           | Iridium      | Ir     | Solid         | 22.42   |
| 78           | Platinum     | Pt     | Solid         | 21.46   |
| 75           | Rhenium      | Re     | Solid         | 20.8    |
| 93           | Neptunium    | Np     | Solid         | 20.25   |
| 94           | Plutonium    | Pu     | Solid         | 19.84   |
| 74           | Tungsten     | W      | Solid         | 19.3    |
| 79           | Gold         | Au     | Solid         | 19.282  |
| 92           | Uranium      | U      | Solid         | 18.95   |
| 73           | Tantalum     | Ta     | Solid         | 16.4    |
| 91           | Protactinium | Pa     | Solid         | 15.37   |
+--------------+--------------+--------+---------------+---------+
sqlite> select Name, Symbol, StandardState, MeltingPoint, BoilingPoint
   ...> from element where BoilingPoint < 300 order by BoilingPoint;
+----------+--------+---------------+--------------+--------------+
|   Name   | Symbol | StandardState | MeltingPoint | BoilingPoint |
+----------+--------+---------------+--------------+--------------+
| Helium   | He     | Gas           | 0.95         | 4.22         |
| Hydrogen | H      | Gas           | 13.81        | 20.28        |
| Neon     | Ne     | Gas           | 24.56        | 27.07        |
| Nitrogen | N      | Gas           | 63.15        | 77.36        |
| Fluorine | F      | Gas           | 53.53        | 85.03        |
| Argon    | Ar     | Gas           | 83.8         | 87.3         |
| Oxygen   | O      | Gas           | 54.36        | 90.2         |
| Krypton  | Kr     | Gas           | 115.79       | 119.93       |
| Xenon    | Xe     | Gas           | 161.36       | 165.03       |
| Radon    | Rn     | Gas           | 202.0        | 211.45       |
| Chlorine | Cl     | Gas           | 171.65       | 239.11       |
+----------+--------+---------------+--------------+--------------+
```

Try following practice.

Practice 06-05

Try a SQL query for the table "`element`".

Save the database into a file.

```
sqlite> .save element.db
```

Quit from SQLite.

```
sqlite> .quit
```

5 Constructing database from Bright Star Catalogue

Download Bright Star Catalogue and construct a database of bright stars.

5.1 Downloading the catalogue

Make a Python script to download Bright Star Catalogue. The URL of the catalogue file is following.

- `https://cdsarc.cds.unistra.fr/ftp/V/50/catalog.gz`

Here is an example of Python script for downloading the file.

Python Code 3: ai202209_s06_02.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 07:28:31 (CST) daisuke>
#

# importing urllib module
import urllib.request

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# URL of data file
url_data = 'https://cdsarc.cds.unistra.fr/ftp/V/50/catalog.gz'

# output file name
file_output = 'bsc.catalog.gz'

# printing status
print (f'Now, fetching {url_data}...')

# opening URL
with urllib.request.urlopen (url_data) as fh_read:
    # reading data
    data_byte = fh_read.read ()

# printing status
print (f'Finished fetching {url_data}!')

# printing status
print (f'Now, writing the data into file "{file_output}"...')

# opening file for writing
with open (file_output, 'wb') as fh_write:
    # writing data
    fh_write.write (data_byte)

# printing status
print (f'Finished writing the data into file "{file_output}"!')
```

Execute above script to download the Bright Star Catalogue.

```
% ./ai202209_s06_02.py
Now, fetching https://cdsarc.cds.unistra.fr/ftp/V/50/catalog.gz...
Finished fetching https://cdsarc.cds.unistra.fr/ftp/V/50/catalog.gz!
Now, writing the data into file "bsc.catalog.gz"...
Finished writing the data into file "bsc.catalog.gz"!
```

Check downloaded file.

```
% ls -lF bsc*
-rw-r--r--  1 daisuke  taiwan  573921 Oct 23 07:28 bsc.catalog.gz
% file bsc.catalog.gz
bsc.catalog.gz: gzip compressed data, was "catalog", last modified: Mon Oct  4 0
9:55:01 1993, max compression, from Unix, original size modulo 2^32 1704879
```

Also, download "ReadMe" file. The "ReadMe" file contains the description about the structure of the catalogue file.

Python Code 4: ai202209_s06_03.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 07:27:00 (CST) daisuke>
#

# importing urllib module
import urllib.request

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# URL of data file
url_data = 'http://cdsarc.u-strasbg.fr/ftp/V/50/ReadMe'

# output file name
file_output = 'bsc.readme'

# printing status
print (f'Now, fetching {url_data}...')

# opening URL
with urllib.request.urlopen (url_data) as fh_read:
    # reading data
    data_byte = fh_read.read ()

# printing status
print (f'Finished fetching {url_data}!')

# converting raw byte data into string
data_str = data_byte.decode ('utf-8')

# printing status
print (f'Now, writing data into file "{file_output}"...')
```

```
# opening file for writing
with open (file_output, 'w') as fh_write:
    # writing data
    fh_write.write (data_str)

# printing status
print (f'Finished writing data into file "{file_output}"!')
```

Execute above script to download "ReadMe" file of the Bright Star Catalogue.

```
% ./ai202209_s06_03.py
Now, fetching http://cdsarc.u-strasbg.fr/ftp/V/50/ReadMe...
Finished fetching http://cdsarc.u-strasbg.fr/ftp/V/50/ReadMe!
Now, writing data into file "bsc.readme"...
Finished writing data into file "bsc.readme"!
```

Check downloaded file.

```
% ls -lF bsc*
-rw-r--r--  1 daisuke  taiwan  573921 Oct 23 07:28 bsc.catalog.gz
-rw-r--r--  1 daisuke  taiwan   11571 Oct 23 07:31 bsc.readme
% file bsc.readme
bsc.readme: ASCII text
% head bsc.readme
V/50           Bright Star Catalogue, 5th Revised Ed.      (Hoffleit+, 1991)
================================================================================
The Bright Star Catalogue,  5th Revised Ed. (Preliminary Version)
     Hoffleit D., Warren Jr W.H.
    <Astronomical Data Center, NSSDC/ADC (1991)>
    =1964BS....C......0H
    =1991bsc..book.....H
================================================================================
ADC_Keywords: Combined data ; Stars, bright
```

Read the byte-by-byte description part of "ReadMe" file to learn about the format of the catalogue file. (Fig. 20)

```
Byte-by-byte Description of file: catalog
--------------------------------------------------------------------------------
   Bytes Format  Units   Label    Explanations
--------------------------------------------------------------------------------
   1-  4  I4     ---     HR       [1/9110]+ Harvard Revised Number
                                    = Bright Star Number
   5- 14  A10    ---     Name     Name, generally Bayer and/or Flamsteed name
  15- 25  A11    ---     DM       Durchmusterung Identification (zone in
                                    bytes 17-19)
  26- 31  I6     ---     HD       [1/225300]? Henry Draper Catalog Number
  32- 37  I6     ---     SAO      [1/258997]? SAO Catalog Number
  38- 41  I4     ---     FK5      ? FK5 star Number
      42  A1     ---     IRflag   [I] I if infrared source
      43  A1     ---   r_IRflag  *[ ':] Coded reference for infrared source
      44  A1     ---    Multiple *[AWDIRS] Double or multiple-star code
  45- 49  A5     ---     ADS      Aitken's Double Star Catalog (ADS) designation
  50- 51  A2     ---     ADScomp  ADS number components
  52- 60  A9     ---     VarID    Variable star identification
  61- 62  I2     h       RAh1900  ?Hours RA, equinox B1900, epoch 1900.0 (1)
  63- 64  I2     min     RAm1900  ?Minutes RA, equinox B1900, epoch 1900.0 (1)
  65- 68  F4.1   s       RAs1900  ?Seconds RA, equinox B1900, epoch 1900.0 (1)
```

```
      69  A1     ---     DE-1900  ?Sign Dec, equinox B1900, epoch 1900.0 (1)
  70- 71  I2     deg     DEd1900  ?Degrees Dec, equinox B1900, epoch 1900.0 (1)
  72- 73  I2     arcmin  DEm1900  ?Minutes Dec, equinox B1900, epoch 1900.0 (1)
  74- 75  I2     arcsec  DEs1900  ?Seconds Dec, equinox B1900, epoch 1900.0 (1)
  76- 77  I2     h       RAh      ?Hours RA, equinox J2000, epoch 2000.0 (1)
  78- 79  I2     min     RAm      ?Minutes RA, equinox J2000, epoch 2000.0 (1)
  80- 83  F4.1   s       RAs      ?Seconds RA, equinox J2000, epoch 2000.0 (1)
      84  A1     ---     DE-      ?Sign Dec, equinox J2000, epoch 2000.0 (1)
  85- 86  I2     deg     DEd      ?Degrees Dec, equinox J2000, epoch 2000.0 (1)
  87- 88  I2     arcmin  DEm      ?Minutes Dec, equinox J2000, epoch 2000.0 (1)
  89- 90  I2     arcsec  DEs      ?Seconds Dec, equinox J2000, epoch 2000.0 (1)
  91- 96  F6.2   deg     GLON     ?Galactic longitude (1)
  97-102  F6.2   deg     GLAT     ?Galactic latitude (1)
 103-107  F5.2   mag     Vmag     ?Visual magnitude (1)
     108  A1     ---   n_Vmag    *[ HR] Visual magnitude code
     109  A1     ---   u_Vmag     [ :?] Uncertainty flag on V
 110-114  F5.2   mag     B-V      ? B-V color in the UBV system
     115  A1     ---   u_B-V      [ :?] Uncertainty flag on B-V
 116-120  F5.2   mag     U-B      ? U-B color in the UBV system
     121  A1     ---   u_U-B      [ :?] Uncertainty flag on U-B
 122-126  F5.2   mag     R-I      ? R-I   in system specified by n_R-I
     127  A1     ---   n_R-I      [CE:?D] Code for R-I system (Cousin, Eggen)
 128-147  A20    ---     SpType   Spectral type
     148  A1     ---   n_SpType   [evt] Spectral type code
 149-154  F6.3 arcsec/yr pmRA    *?Annual proper motion in RA J2000, FK5 system
 155-160  F6.3 arcsec/yr pmDE     ?Annual proper motion in Dec J2000, FK5 system
     161  A1     ---   n_Parallax [D] D indicates a dynamical parallax,
                                    otherwise a trigonometric parallax
 162-166  F5.3   arcsec  Parallax ? Trigonometric parallax (unless n_Parallax)
 167-170  I4     km/s    RadVel   ? Heliocentric Radial Velocity
 171-174  A4     ---   n_RadVel  *[V?SB123O ] Radial velocity comments
 175-176  A2     ---   l_RotVel   [<=> ] Rotational velocity limit characters
 177-179  I3     km/s    RotVel   ? Rotational velocity, v sin i
     180  A1     ---   u_RotVel   [ :v] uncertainty and variability flag on
                                    RotVel
 181-184  F4.1   mag     Dmag     ? Magnitude difference of double,
                                    or brightest multiple
 185-190  F6.1   arcsec  Sep      ? Separation of components in Dmag
                                    if occultation binary.
 191-194  A4     ---     MultID   Identifications of components in Dmag
 195-196  I2     ---     MultCnt  ? Number of components assigned to a multiple
     197  A1     ---     NoteFlag [*] a star indicates that there is a note
                                    (see file notes)
--------------------------------------------------------------------------------
```

5.2 Reading Bright Star Catalogue

Make a Python script to open and read the Bright Star Catalogue. Here is a sample Python script to read HR number, name of star, RA, Dec, galactic longitude, galactic latitude, V-band magnitude, (B-V) colour index, spectral type, proper motion, and parallax from Bright Star Catalogue.

Python Code 5: ai202209_s06_04.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 14:00:13 (CST) daisuke>
#
```

```
koenji_20221023_073925
Byte-by-byte Description of file: catalog
--------------------------------------------------------------------------------
   Bytes Format  Units   Label     Explanations
--------------------------------------------------------------------------------
   1-  4  I4     ---     HR        [1/9110]+ Harvard Revised Number
                                     = Bright Star Number
   5- 14  A10    ---     Name      Name, generally Bayer and/or Flamsteed name
  15- 25  A11    ---     DM        Durchmusterung Identification (zone in
                                     bytes 17-19)
  26- 31  I6     ---     HD        [1/225300]? Henry Draper Catalog Number
  32- 37  I6     ---     SAO       [1/258997]? SAO Catalog Number
  38- 41  I4     ---     FK5       ? FK5 star Number
      42  A1     ---     IRflag    [I] I if infrared source
      43  A1     ---   r_IRflag  *[ ':] Coded reference for infrared source
      44  A1     ---    Multiple *[AWDIRS] Double or multiple-star code
  45- 49  A5     ---     ADS       Aitken's Double Star Catalog (ADS) designation
  50- 51  A2     ---     ADScomp   ADS number components
  52- 60  A9     ---     VarID     Variable star identification
  61- 62  I2     h       RAh1900   ?Hours RA, equinox B1900, epoch 1900.0 (1)
  63- 64  I2     min     RAm1900   ?Minutes RA, equinox B1900, epoch 1900.0 (1)
  65- 68  F4.1   s       RAs1900   ?Seconds RA, equinox B1900, epoch 1900.0 (1)
      69  A1     ---     DE-1900   ?Sign Dec, equinox B1900, epoch 1900.0 (1)
  70- 71  I2     deg     DEd1900   ?Degrees Dec, equinox B1900, epoch 1900.0 (1)
  72- 73  I2     arcmin  DEm1900   ?Minutes Dec, equinox B1900, epoch 1900.0 (1)
  74- 75  I2     arcsec  DEs1900   ?Seconds Dec, equinox B1900, epoch 1900.0 (1)
  76- 77  I2     h       RAh       ?Hours RA, equinox J2000, epoch 2000.0 (1)
:
```

Figure 20: The byte-by-byte description part of the "ReadMe" file.

```python
# importing gzip module
import gzip

# importing sys module
import sys

# catalogue file name
file_catalogue = 'bsc.catalog.gz'

# opening catalogue file
with gzip.open (file_catalogue, 'rb') as fh:
    # reading catalogue line-by-line
    for line in fh:
        # Harvard Revised Number of star
        try:
            HR = int (line[0:4])
        except:
            # printing message
            print (f'Something is wrong with following line...')
            print (f'  {line[:75]}')
            print (f'Cannot extract HR number!')
            # exit
            sys.exit (1)
        # name of star
        name = line[4:14].strip ().decode ('utf-8')
        if (name == ''):
            name = '__NONE__'
        # RA
        try:
            RA_h = int (line[75:77])
```

```
            RA_m = int (line[77:79])
            RA_s = float (line[79:83])
        except:
            RA_h = 99
            RA_m = 99
            RA_s = 99.9
        RA_str = f'{RA_h:02d}:{RA_m:02d}:{RA_s:04.1f}'
        RA_deg = (RA_h + RA_m / 60.0 + RA_s / 3600.0) * 15.0
        # Dec
        try:
            Dec_sign = line[83:84].decode ('utf-8')
            Dec_d    = int (line[84:86])
            Dec_m    = int (line[86:88])
            Dec_s    = int (line[88:90])
        except:
            Dec_sign = '-'
            Dec_d    = 99
            Dec_m    = 99
            Dec_s    = 99
        Dec_str = f'{Dec_sign}{Dec_d:02d}:{Dec_m:02d}:{Dec_s:02d}'
        if (Dec_sign == '+'):
            Dec_deg = Dec_d + Dec_m / 60.0 + Dec_s / 3600.0
        else:
            Dec_deg = (Dec_d + Dec_m / 60.0 + Dec_s / 3600.0) * (-1.0)
        # galactic longitude
        try:
            glon = float (line[90:96])
        except:
            glon = -999.99
        # galactic latitude
        try:
            glat = float (line[96:102])
        except:
            glat = -999.99
        # Vmag
        try:
            mag_V = float (line[102:107])
        except:
            mag_V = -999.9
        # B-V colour
        try:
            colour_BV = float (line[109:114])
        except:
            colour_BV = -999.9
        # spectral type
        sptype = line[127:147].strip ().decode ('utf-8')
        # proper motion RA
        try:
            pm_RA = float (line[148:154])
        except:
            pm_RA = -999.9
        # proper motion Dec
        try:
            pm_Dec = float (line[154:160])
        except:
            pm_Dec = -999.9
        # parallax
        try:
            parallax = float (line[161:166])
```

```
        except:
            parallax = -999.9


        # printing extracted data
        print (f'HR = {HR}')
        print (f'  name     = "{name}"')
        print (f'  RA_str   = {RA_str}')
        print (f'  RA_deg   = {RA_deg}')
        print (f'  Dec_str  = {Dec_str}')
        print (f'  Dec_deg  = {Dec_deg}')
        print (f'  glon     = {glon}')
        print (f'  glat     = {glat}')
        print (f'  Vmag     = {mag_V}')
        print (f'  B-V      = {colour_BV}')
        print (f'  sptype   = "{sptype}"')
        print (f'  pmRA     = {pm_RA}')
        print (f'  pmDec    = {pm_Dec}')
        print (f'  parallax = {parallax}')
```

Execute above script to read Bright Star Catalogue.

```
% chmod a+x ai202209_s06_04.py
% ./ai202209_s06_04.py > bsc_extracted.data
% file bsc_extracted.data
bsc_extracted.data: ASCII text
% ls -lF bsc_extracted.data
-rw-r--r--  1 daisuke  taiwan  2694265 Oct 23 13:23 bsc_extracted.data
% head -20 bsc_extracted.data
HR = 1
  name     = "__NONE__"
  RA_str   = 00:05:09.9
  RA_deg   = 1.29125
  Dec_str  = +45:13:45
  Dec_deg  = 45.22916666666667
  glon     = 114.44
  glat     = -16.88
  Vmag     = 6.7
  B-V      = 0.07
  sptype   = "A1Vn"
  pmRA     = -0.012
  pmDec    = -0.018
  parallax = -999.9
HR = 2
  name     = "__NONE__"
  RA_str   = 00:05:03.8
  RA_deg   = 1.2658333333333334
  Dec_str  = -00:30:11
  Dec_deg  = -0.5030555555555556
```

5.3 Making a SQL file to generate a database

Make a Python script to read the Bright Star Catalogue, and generate a SQL file to construct a database. Here is an example.

Python Code 6: ai202209_s06_05.py

```
#!/usr/pkg/bin/python3.9
```

```
#
# Time-stamp: <2022/10/23 14:26:14 (CST) daisuke>
#

# importing gzip module
import gzip

# importing sys module
import sys

# catalogue file name
file_catalogue = 'bsc.catalog.gz'

# SQL file name
file_sql = 'bsc_makedb.sql'

# opening file for writing
with open (file_sql, 'w') as fh_sql:
    # SQL command to create a table
    sql_table = f'create table bsc (hr integer primary key, name text, ' \
        + f'ra_str text, ra_deg real, dec_str text, dec_deg real, ' \
        + f'glon real, glat real, vmag real, bv real, sptype text, ' \
        + f'pmra real, pmdec real, parallax real);\n'
    fh_sql.write (sql_table)

    # opening catalogue file
    with gzip.open (file_catalogue, 'rb') as fh_bsc:
        # reading catalogue line-by-line
        for line in fh_bsc:
            # Harvard Revised Number of star
            try:
                HR = int (line[0:4])
            except:
                # printing message
                print (f'Something is wrong with following line...')
                print (f'  {line[:75]}')
                print (f'Cannot extract HR number!')
                # exit
                sys.exit (1)
            # name of star
            name = line[4:14].strip ().decode ('utf-8')
            if (name == ''):
                name = '__NONE__'
            # RA
            try:
                RA_h = int (line[75:77])
                RA_m = int (line[77:79])
                RA_s = float (line[79:83])
            except:
                RA_h = 99
                RA_m = 99
                RA_s = 99.9
            RA_str = f'{RA_h:02d}:{RA_m:02d}:{RA_s:04.1f}'
            RA_deg = (RA_h + RA_m / 60.0 + RA_s / 3600.0) * 15.0
            # Dec
            try:
                Dec_sign = line[83:84].decode ('utf-8')
                Dec_d    = int (line[84:86])
                Dec_m    = int (line[86:88])
```

```
                Dec_s    = int (line[88:90])
            except:
                Dec_sign = '-'
                Dec_d    = 99
                Dec_m    = 99
                Dec_s    = 99
            Dec_str = f'{Dec_sign}{Dec_d:02d}:{Dec_m:02d}:{Dec_s:02d}'
            if (Dec_sign == '+'):
                Dec_deg = Dec_d + Dec_m / 60.0 + Dec_s / 3600.0
            else:
                Dec_deg = (Dec_d + Dec_m / 60.0 + Dec_s / 3600.0) * (-1.0)
            # galactic longitude
            try:
                glon = float (line[90:96])
            except:
                glon = -999.99
            # galactic latitude
            try:
                glat = float (line[96:102])
            except:
                glat = -999.99
            # Vmag
            try:
                mag_V = float (line[102:107])
            except:
                mag_V = -999.9
            # B-V colour
            try:
                colour_BV = float (line[109:114])
            except:
                colour_BV = -999.9
            # spectral type
            sptype = line[127:147].strip ().decode ('utf-8')
            # proper motion RA
            try:
                pm_RA = float (line[148:154])
            except:
                pm_RA = -999.9
            # proper motion Dec
            try:
                pm_Dec = float (line[154:160])
            except:
                pm_Dec = -999.9
            # parallax
            try:
                parallax = float (line[161:166])
            except:
                parallax = -999.9

            # SQL command to add data to table
            sql_add = f'insert into bsc values ({HR}, "{name}", ' \
                + f'"{RA_str}", {RA_deg}, "{Dec_str}", {Dec_deg}, ' \
                + f'{glon}, {glat}, {mag_V}, {colour_BV}, ' \
                + f'"{sptype}", {pm_RA}, {pm_Dec}, {parallax});\n'
            fh_sql.write (sql_add)
```

Execute above script and make a SQL file.

```
% chmod a+x ai202209_s06_05.py
```

```
% ./ai202209_s06_05.py
% ls -lF *.sql
-rw-r--r--  1 daisuke  taiwan  1501084 Oct 23 14:26 bsc_makedb.sql
-rw-r--r--  1 daisuke  taiwan     3253 Oct 21 15:06 dwarf_planet.sql
% head bsc_makedb.sql | cut -b 1-80
create table bsc (hr integer primary key, name text, ra_str text, ra_deg real, d
insert into bsc values (1, "__NONE__", "00:05:09.9", 1.29125, "+45:13:45", 45.22
insert into bsc values (2, "__NONE__", "00:05:03.8", 1.2658333333333334, "-00:30
insert into bsc values (3, "33    Psc", "00:05:20.1", 1.3337499999999998, "-05:4
insert into bsc values (4, "86    Peg", "00:05:42.0", 1.425, "+13:23:46", 13.396
insert into bsc values (5, "__NONE__", "00:06:16.0", 1.5666666666666669, "+58:26
insert into bsc values (6, "__NONE__", "00:06:19.0", 1.5791666666666668, "-49:04
insert into bsc values (7, "10    Cas", "00:06:26.5", 1.6104166666666668, "+64:1
insert into bsc values (8, "__NONE__", "00:06:36.8", 1.6533333333333333, "+29:01
insert into bsc values (9, "__NONE__", "00:06:50.1", 1.70875, "-23:06:27", -23.1
```

5.4 Making BSC database

Start SQLite and read SQL file.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .read bsc_makedb.sql
sqlite> .tables
bsc
sqlite> .schema --indent bsc
CREATE TABLE bsc(
  hr integer primary key,
  name text,
  ra_str text,
  ra_deg real,
  dec_str text,
  dec_deg real,
  glon real,
  glat real,
  vmag real,
  bv real,
  sptype text,
  pmra real,
  pmdec real,
  parallax real
);
```

5.5 Trying some SQL queries

Try some SQL query for BSC database. Find stars with V-band magnitude brighter than 1.5 mag.

```
sqlite> select hr, name, ra_str, dec_str, vmag, sptype from bsc
   ...> where (vmag < 1.5) and (vmag > -30) order by vmag;
+------+-----------+------------+-----------+-------+-----------------+
|  hr  |   name    |   ra_str   |  dec_str  | vmag  |     sptype      |
+------+-----------+------------+-----------+-------+-----------------+
| 2491 | 9Alp CMa  | 06:45:08.9 | -16:42:58 | -1.46 | A1Vm            |
| 2326 | Alp Car   | 06:23:57.1 | -52:41:45 | -0.72 | F0II            |
```

```
| 5340 | 16Alp Boo | 14:15:39.7 | +19:10:57 | -0.04 | K1.5IIIFe-0.5   |
| 5459 | Alp1Cen   | 14:39:35.9 | -60:50:07 | -0.01 | G2V             |
| 7001 | 3Alp Lyr  | 18:36:56.3 | +38:47:01 | 0.03  | A0Va            |
| 1708 | 13Alp Aur | 05:16:41.4 | +45:59:53 | 0.08  | G5IIIe+G0III    |
| 1713 | 19Bet Ori | 05:14:32.3 | -08:12:06 | 0.12  | B8Ia:           |
| 2943 | 10Alp CMi | 07:39:18.1 | +05:13:30 | 0.38  | F5IV-V          |
| 472  | Alp Eri   | 01:37:42.9 | -57:14:12 | 0.46  | B3Vpe           |
| 2061 | 58Alp Ori | 05:55:10.3 | +07:24:25 | 0.5   | M1-2Ia-Iab      |
| 5267 | Bet Cen   | 14:03:49.4 | -60:22:23 | 0.61  | B1III           |
| 7557 | 53Alp Aql | 19:50:47.0 | +08:52:06 | 0.77  | A7V             |
| 1457 | 87Alp Tau | 04:35:55.2 | +16:30:33 | 0.85  | K5+III          |
| 6134 | 21Alp Sco | 16:29:24.4 | -26:25:55 | 0.96  | M1.5Iab-Ib+B4Ve |
| 5056 | 67Alp Vir | 13:25:11.6 | -11:09:41 | 0.98  | B1III-IV+B2V    |
| 2990 | 78Bet Gem | 07:45:18.9 | +28:01:34 | 1.14  | K0IIIb          |
| 8728 | 24Alp PsA | 22:57:39.1 | -29:37:20 | 1.16  | A3V             |
| 4853 | Bet Cru   | 12:47:43.2 | -59:41:19 | 1.25  | B0.5III         |
| 7924 | 50Alp Cyg | 20:41:25.9 | +45:16:49 | 1.25  | A2Ia            |
| 4730 | Alp1Cru   | 12:26:35.9 | -63:05:57 | 1.33  | B0.5IV          |
| 5460 | Alp2Cen   | 14:39:36.1 | -60:50:08 | 1.33  | K1V             |
| 3982 | 32Alp Leo | 10:08:22.3 | +11:58:02 | 1.35  | B7V             |
+------+-----------+------------+-----------+-------+-----------------+
```

Find stars in the solar neighbourhood.

```
sqlite> select hr, name, ra_str, dec_str, vmag, bv, parallax from bsc
   ...> where parallax >= 0.2 order by parallax desc;
+------+-----------+------------+-----------+-------+------+----------+
|  hr  |   name    |   ra_str   |  dec_str  | vmag  |  bv  | parallax |
+------+-----------+------------+-----------+-------+------+----------+
| 5459 | Alp1Cen   | 14:39:35.9 | -60:50:07 | -0.01 | 0.71 | 0.751    |
| 5460 | Alp2Cen   | 14:39:36.1 | -60:50:08 | 1.33  | 0.88 | 0.751    |
| 2491 | 9Alp CMa  | 06:45:08.9 | -16:42:58 | -1.46 | 0.0  | 0.375    |
| 1084 | 18Eps Eri | 03:32:55.8 | -09:27:30 | 3.73  | 0.88 | 0.303    |
| 8086 | 61    Cyg | 21:06:55.3 | +38:44:36 | 6.03  | 1.37 | 0.294    |
| 8085 | 61    Cyg | 21:06:54.6 | +38:44:45 | 5.21  | 1.18 | 0.292    |
| 2943 | 10Alp CMi | 07:39:18.1 | +05:13:30 | 0.38  | 0.42 | 0.288    |
| 8387 | Eps Ind   | 22:03:21.6 | -56:47:10 | 4.69  | 1.06 | 0.285    |
| 509  | 52Tau Cet | 01:44:04.1 | -15:56:15 | 3.5   | 0.72 | 0.275    |
| 1325 | 40Omi2Eri | 04:15:16.3 | -07:39:10 | 4.43  | 0.82 | 0.209    |
| 6752 | 70    Oph | 18:05:27.3 | +02:29:58 | 4.03  | 0.86 | 0.201    |
+------+-----------+------------+-----------+-------+------+----------+
```

Find stars near the north galactic pole.

```
sqlite> select hr, name, ra_str, dec_str, glon, glat, vmag from bsc
   ...> where glat > 85.0 order by glat desc;
+------+-----------+------------+-----------+--------+-------+------+
|  hr  |   name    |   ra_str   |  dec_str  |  glon  | glat  | vmag |
+------+-----------+------------+-----------+--------+-------+------+
| 4883 | 31    Com | 12:51:41.9 | +27:32:26 | 114.93 | 89.58 | 4.94 |
| 4869 | 30    Com | 12:49:17.4 | +27:33:08 | 171.1  | 89.36 | 5.78 |
| 4864 | __NONE__  | 12:48:47.0 | +24:50:25 | 288.28 | 87.64 | 6.31 |
| 4954 | 41    Com | 13:07:10.7 | +27:37:29 | 41.94  | 86.47 | 4.8  |
| 4956 | __NONE__  | 13:07:53.6 | +27:33:21 | 40.56  | 86.32 | 6.19 |
| 4948 | __NONE__  | 13:06:10.2 | +29:01:46 | 64.11  | 86.23 | 6.54 |
| 4924 | 37    Com | 13:00:16.5 | +30:47:06 | 95.6   | 85.86 | 4.9  |
| 4873 | __NONE__  | 12:50:17.4 | +22:51:48 | 299.36 | 85.73 | 6.43 |
```

```
| 4983 | 43Bet Com | 13:11:52.4 | +27:52:41 | 43.33  | 85.4  | 4.26 |
| 4780 | 22    Com | 12:33:34.2 | +24:16:59 | 247.2  | 85.07 | 6.29 |
+------+-----------+------------+-----------+--------+-------+------+
```

Find O-type stars.

```
sqlite> select hr, name, glon, glat, vmag, bv, sptype from bsc
   ...> where sptype like '%O%' order by glat;
+------+-----------+--------+--------+------+-------+-----------------+
|  hr  |   name    |  glon  |  glat  | vmag |  bv   |     sptype      |
+------+-----------+--------+--------+------+-------+-----------------+
| 2212 | Del Pic   | 263.3  | -27.68 | 4.81 | -0.23 | B3III+O9V       |
| 1996 | Mu  Col   | 237.29 | -27.1  | 5.17 | -0.28 | O9.5V           |
| 1899 | 44Iot Ori | 209.52 | -19.58 | 2.77 | -0.24 | O9III           |
| 1895 | 41The1Ori | 209.01 | -19.38 | 5.13 | 0.02  | O6p             |
| 1897 | 43The2Ori | 209.05 | -19.37 | 5.08 | -0.09 | O9.5Vp          |
| 1852 | 34Del Ori | 203.86 | -17.74 | 2.23 | -0.22 | O9.5II          |
| 1931 | 48Sig Ori | 206.82 | -17.34 | 3.81 | -0.24 | O9.5V           |
| 1209 | __NONE__  | 163.08 | -17.14 | 6.1  | 0.29  | O9.5ep          |
| 8622 | 10    Lac | 96.65  | -16.98 | 4.88 | -0.2  | O9V             |
| 1948 | 50Zet Ori | 206.45 | -16.59 | 2.05 | -0.21 | O9.7Ib          |
| 1228 | 46Xi  Per | 160.37 | -13.11 | 4.04 | 0.01  | O7.5III(n)((f)) |
| 1879 | 39Lam Ori | 195.05 | -12.0  | 3.54 | -0.18 | O8III((f))      |
| 65   | __NONE__  | 117.59 | -11.09 | 6.14 | -0.13 | O9IIInn         |
| 3207 | Gam2Vel   | 262.8  | -7.69  | 1.78 | -0.22 | WC8+O9I         |
| 2782 | 30Tau CMa | 238.18 | -5.54  | 4.4  | -0.15 | O9Ib            |
| 2781 | 29    CMa | 237.82 | -5.37  | 4.98 | -0.15 | O7Ia:fp         |
| 3165 | Zet Pup   | 255.98 | -4.71  | 2.25 | -0.26 | O5f             |
| 7574 | 9     Sge | 56.48  | -4.33  | 6.23 | 0.01  | O7.5Iaf         |
| 8154 | 68    Cyg | 87.61  | -3.84  | 5.0  | -0.01 | O7.5III:n((f))  |
| 5664 | Del Cir   | 319.69 | -2.91  | 5.09 | -0.06 | O8.5V           |
| 5680 | __NONE__  | 320.13 | -2.64  | 5.46 | -0.1  | O7.5III((f))    |
| 2679 | __NONE__  | 225.68 | -2.32  | 6.48 | -0.1  | O7.5V           |
| 1712 | __NONE__  | 172.08 | -2.26  | 5.96 | 0.22  | O9.5V           |
| 2442 | __NONE__  | 210.03 | -2.11  | 6.21 | 0.15  | O9.5II          |
| 3219 | __NONE__  | 254.47 | -2.02  | 6.44 | -0.01 | O9.5II          |
| 6823 | 16    Sgr | 10.76  | -1.58  | 5.95 | 0.02  | O9.5II          |
| 6187 | __NONE__  | 336.71 | -1.57  | 5.65 | 0.13  | O5III(f)        |
| 6736 | 9     Sgr | 6.01   | -1.2   | 5.97 | 0.0   | O4V((f))        |
| 6841 | __NONE__  | 12.7   | -1.13  | 6.54 | 0.11  | O7III:(n)((f))  |
| 2694 | __NONE__  | 224.17 | -0.78  | 6.21 | 0.03  | O6.5V           |
| 2422 | __NONE__  | 205.87 | -0.31  | 6.06 | 0.05  | O8p             |
| 8023 | __NONE__  | 85.7   | -0.3   | 5.96 | 0.05  | O6V((f))        |
| 6535 | __NONE__  | 355.67 | 0.05   | 5.7  | 0.04  | O7V+O7V         |
| 6672 | __NONE__  | 4.54   | 0.3    | 6.2  | 0.04  | O7.5II((f))     |
| 2467 | __NONE__  | 206.21 | 0.8    | 6.37 | -0.05 | O6.5V           |
| 6263 | __NONE__  | 343.45 | 1.16   | 6.45 | 0.2   | O9Ib            |
| 6265 | __NONE__  | 343.49 | 1.16   | 6.59 | 0.21  | WC7+O5-8        |
| 6272 | __NONE__  | 344.08 | 1.49   | 5.77 | 0.15  | O8:Iafpe        |
| 6245 | __NONE__  | 343.62 | 1.94   | 5.22 | 0.07  | O8Iaf           |
| 8406 | 14    Cep | 102.01 | 2.18   | 5.56 | 0.06  | O9Vn            |
| 2456 | 15    Mon | 202.94 | 2.2    | 4.66 | -0.25 | O7V((f))        |
| 8469 | 22Lam Cep | 103.83 | 2.61   | 5.04 | 0.25  | O6I(n)fp        |
| 2806 | __NONE__  | 224.41 | 2.63   | 6.43 | -0.19 | O9V             |
| 7767 | __NONE__  | 78.1   | 2.78   | 5.84 | 0.1   | O9V             |
| 6397 | __NONE__  | 352.59 | 2.87   | 5.53 | -0.01 | O7.5V[n]e       |
| 6164 | __NONE__  | 340.54 | 3.01   | 5.47 | 0.4   | O9Ia            |
| 6347 | __NONE__  | 349.97 | 3.22   | 6.13 | 0.5   | O9.5Iab         |
```

```
| 8281 | __NONE__  | 99.29  | 3.74   | 5.62 | 0.21  | 06.5V((f))      |
| 8428 | 19    Cep | 104.87 | 5.39   | 5.11 | 0.08  | 09.5Ib          |
| 4908 | __NONE__  | 303.55 | 6.03   | 5.32 | 0.01  | 09Ib            |
| 8327 | __NONE__  | 103.14 | 6.99   | 5.95 | 0.31  | 09Ib-II         |
| 7589 | __NONE__  | 80.99  | 10.09  | 5.62 | -0.07 | 09.5Ia          |
| 1542 | 9Alp Cam  | 144.07 | 14.04  | 4.29 | 0.03  | 09.5Ia          |
| 6765 | 98    Her | 48.53  | 19.55  | 5.06 | 1.58  | M3-IIIZrO 0+    |
| 6175 | 13Zet Oph | 6.28   | 23.59  | 2.56 | 0.02  | 09.5Vn          |
+------+-----------+--------+--------+------+-------+-----------------+
```

Try following practice.

Practice 06-06

Try 3 SQL queries for the table "`bsc`".

5.6 Saving the database into a file

Save the database into a file.

```
sqlite> .save bsc.db
```

Quit from SQLite.

```
sqlite> .quit
```

Now, you have a database file named "`bsc.db`".

```
% ls -lF bsc.db
-rw-r--r--  1 daisuke  taiwan  1146880 Oct 23 14:39 bsc.db
% file bsc.db
bsc.db: SQLite 3.x database, last written using SQLite version 3039004, file cou
nter 1, database pages 280, cookie 0x1, schema 4, UTF-8, version-valid-for 1
```

6 Making a database from Hipparchos catalogue

Download Hipparchos catalogue and construct a database from it.

6.1 Downloading Hipparcos catalogue

Make a Python script to download Hipparcos catalogue. Here is an example.

Python Code 7: ai202209_s06_06.py

```python
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 17:55:02 (CST) daisuke>
#

# importing urllib module
import urllib.request

# importing ssl module
import ssl

```

```
# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# URL of data file
url_data = 'https://cdsarc.cds.unistra.fr/ftp/I/239/hip_main.dat'

# output file name
file_output = 'hip_main.dat'

# printing status
print (f'Now, fetching {url_data}...')

# opening URL
with urllib.request.urlopen (url_data) as fh_read:
    # reading data
    data_byte = fh_read.read ()

# printing status
print (f'Finished fetching {url_data}!')

# converting raw byte data into string
data_str = data_byte.decode ('utf-8')

# printing status
print (f'Now, writing data into file "{file_output}"...')

# opening file for writing
with open (file_output, 'w') as fh_write:
    # writing data
    fh_write.write (data_str)

# printing status
print (f'Finished writing data into file "{file_output}"!')
```

Execute above script and download Hipparcos catalogue. It may take a few minutes.

```
% chmod a+x ai202209_s06_06.py
% ./ai202209_s06_06.py
Now, fetching https://cdsarc.cds.unistra.fr/ftp/I/239/hip_main.dat...
Finished fetching https://cdsarc.cds.unistra.fr/ftp/I/239/hip_main.dat!
Now, writing data into file "hip_main.dat"...
Finished writing data into file "hip_main.dat"!
% ls -lF hip_main.dat
-rw-r--r--  1 daisuke  taiwan  53316318 Oct 23 17:56 hip_main.dat
% head hip_main.dat | cut -b 1-80
H|           1| |00 00 00.22|+01 05 20.4| 9.10| |H|000.00091185|+01.08901332| |
H|           2| |00 00 00.91|-19 29 55.8| 9.27| |G|000.00379737|-19.49883745|+|
H|           3| |00 00 01.20|+38 51 33.4| 6.61| |G|000.00500795|+38.85928608| |
H|           4| |00 00 02.01|-51 53 36.8| 8.06| |H|000.00838170|-51.89354612| |
H|           5| |00 00 02.39|-40 35 28.4| 8.55| |H|000.00996534|-40.59122440| |
H|           6| |00 00 04.35|+03 56 47.4|12.31| |G|000.01814144|+03.94648893| |
H|           7| |00 00 05.41|+20 02 11.8| 9.64| |G|000.02254891|+20.03660216| |
H|           8| |00 00 06.55|+25 53 11.3| 9.05|3|H|000.02729160|+25.88647445| |
H|           9| |00 00 08.48|+36 35 09.4| 8.59| |H|000.03534189|+36.58593777| |
H|          10| |00 00 08.70|-50 52 01.5| 8.59| |H|000.03625309|-50.86707360| |
```

Also, download “`ReadMe`” file. Here is a sample Python script.

Python Code 8: ai202209_s06_07.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 18:00:15 (CST) daisuke>
#

# importing urllib module
import urllib.request

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# URL of data file
url_data = 'https://cdsarc.cds.unistra.fr/ftp/I/239/ReadMe'

# output file name
file_output = 'hip_main.readme'

# printing status
print (f'Now, fetching {url_data}...')

# opening URL
with urllib.request.urlopen (url_data) as fh_read:
    # reading data
    data_byte = fh_read.read ()

# printing status
print (f'Finished fetching {url_data}!')

# converting raw byte data into string
data_str = data_byte.decode ('utf-8')

# printing status
print (f'Now, writing data into file "{file_output}"...')

# opening file for writing
with open (file_output, 'w') as fh_write:
    # writing data
    fh_write.write (data_str)

# printing status
print (f'Finished writing data into file "{file_output}"!')
```

Execute above script.

```
% ./ai202209_s06_07.py
Now, fetching https://cdsarc.cds.unistra.fr/ftp/I/239/ReadMe...
Finished fetching https://cdsarc.cds.unistra.fr/ftp/I/239/ReadMe!
Now, writing data into file "hip_main.readme"...
Finished writing data into file "hip_main.readme"!
% ls -lF hip_main.*
-rw-r--r--  1 daisuke  taiwan  53316318 Oct 23 17:56 hip_main.dat
-rw-r--r--  1 daisuke  taiwan     69019 Oct 23 18:01 hip_main.readme
% head hip_main.readme
I/239           The Hipparcos and Tycho Catalogues                      (ESA 1997)
```

```
================================================================================
The Hipparcos and Tycho Catalogues
    ESA 1997
   <ESA, 1997, The Hipparcos Catalogue, ESA SP-1200>
   <ESA, 1997, The Tycho Catalogue, ESA SP-1200>
   =1997HIP...C......0E
================================================================================
ADC_Keywords: Positional data ; Proper motions ; Parallaxes, trigonometric ;
              Photometry ; Fundamental catalog ; Stars, double and multiple
```

6.2 Reading Hipparcos catalogue

Make a Python script to read Hipparchos catalogue. Here is an example.

Python Code 9: ai202209_s06_08.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 19:37:12 (CST) daisuke>
#

# importing sys module
import sys

# catalogue file name
file_catalogue = 'hip_main.dat'

# opening catalogue file
with open (file_catalogue, 'r') as fh_hip:
    # reading catalogue line-by-line
    for line in fh_hip:
        # Hipparcos Number of star
        try:
            hip = int (line[8:14])
        except:
            # printing message
            print (f'Something is wrong with following line...')
            print (f'  {line[:75]}')
            print (f'Cannot extract Hipparcos number!')
            # exit
            sys.exit (1)
        # RA in hhmmss format
        try:
            RA_hms = line[17:28].strip ()
        except:
            RA_hms = '99 99 99.99'
        # Dec in ddmmss format
        try:
            Dec_dms = line[29:40].strip ()
        except:
            Dec_dms = '-99 99 99.9'
        # V-band magnitude
        try:
            mag_V = float (line[41:46])
        except:
            mag_V = -99.99
        # RA in deg
        try:
```

```
            RA_deg = float (line[51:63])
        except:
            RA_deg = -999.99
        # Dec in deg
        try:
            Dec_deg = float (line[64:76])
        except:
            Dec_deg = -999.99
        # parallax in mas
        try:
            parallax = float (line[79:86])
        except:
            parallax = -999999.99
        # proper motion in RA
        try:
            pm_RA = float (line[87:95])
        except:
            pm_RA = -999999.99
        # proper motion in Dec
        try:
            pm_Dec = float (line[96:104])
        except:
            pm_Dec = -999999.99
        # (B-V) colour index
        try:
            colour_BV = float (line[245:251])
        except:
            colour_BV = -999.99
        # (V-I) colour index
        try:
            colour_VI = float (line[260:264])
        except:
            colour_VI = -999.99
        # spectral type
        try:
            sptype = line[435:447].strip ()
        except:
            sptype = '___NONE___'

        # printing extracted data
        print (f'HIP = {hip}')
        print (f'  RA_hms   = "{RA_hms}"')
        print (f'  RA_deg   = {RA_deg}')
        print (f'  Dec_dms  = "{Dec_dms}"')
        print (f'  Dec_deg  = {Dec_deg}')
        print (f'  Vmag     = {mag_V}')
        print (f'  B-V      = {colour_BV}')
        print (f'  V-I      = {colour_VI}')
        print (f'  parallax = {parallax}')
        print (f'  pmRA     = {pm_RA}')
        print (f'  pmDec    = {pm_Dec}')
        print (f'  sptype   = "{sptype}"')
```

Execute above script to open and read Hipparcos catalogue.

```
% chmod a+x ai202209_s06_08.py
% ./ai202209_s06_08.py > hip_main.txt
% ls -lF hip_main.*
-rw-r--r--  1 daisuke  taiwan  53316318 Oct 23 17:56 hip_main.dat
```

```
-rw-r--r--  1 daisuke  taiwan     69019 Oct 23 18:01 hip_main.readme
-rw-r--r--  1 daisuke  taiwan  29271252 Oct 23 19:38 hip_main.txt
% head -20 hip_main.txt
HIP = 1
  RA_hms   = "00 00 00.22"
  RA_deg   = 0.00091185
  Dec_dms  = "+01 05 20.4"
  Dec_deg  = 1.08901332
  Vmag     = 9.1
  B-V      = 0.482
  V-I      = 0.55
  parallax = 3.54
  pmRA     = -5.2
  pmDec    = -1.88
  sptype   = "F5"
HIP = 2
  RA_hms   = "00 00 00.91"
  RA_deg   = 0.00379737
  Dec_dms  = "-19 29 55.8"
  Dec_deg  = -19.49883745
  Vmag     = 9.27
  B-V      = 0.999
  V-I      = 1.04
```

6.3 Constructing SQLite database table using Python script

Make a SQLite database table using Python script. Here is an example.

Python Code 10: ai202209_s06_09.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 18:52:42 (CST) daisuke>
#

# importing sqlite module
import sqlite3

# database file name
file_db = 'hip.db'

# SQL command for making a table
sql_maketable = f'create table hip (hip integer primary key, ' \
    + f'ra_hms text, ra_deg real, dec_dms text, dec_deg real, ' \
    + f'vmag real, bv real, vi real, parallax real, ' \
    + f'pmra real, pmdec real, sptype text);'

# connecting to database
conn   = sqlite3.connect (file_db)
cursor = conn.cursor ()

# making a table
cursor.execute (sql_maketable)

# committing transaction
conn.commit ()

# closing connection
```

```
conn.close ()
```

Execute above script to make a table “hip”.

```
% chmod a+x ai202209_s06_09.py
% ./ai202209_s06_09.py
% ls -lF hip*
-rw-r--r--  1 daisuke  taiwan      8192 Oct 23 19:42 hip.db
-rw-r--r--  1 daisuke  taiwan  53316318 Oct 23 17:56 hip_main.dat
-rw-r--r--  1 daisuke  taiwan     69019 Oct 23 18:01 hip_main.readme
-rw-r--r--  1 daisuke  taiwan  29271252 Oct 23 19:38 hip_main.txt
% file hip.db
hip.db: SQLite 3.x database, last written using SQLite version 3026000, file cou
nter 1, database pages 2, cookie 0x1, schema 4, UTF-8, version-valid-for 1
```

Use SQLite command-line program to check the database file “hip.db”.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open hip.db
sqlite> .tables
hip
sqlite> .schema --indent hip
CREATE TABLE hip(
  hip integer primary key,
  ra_hms text,
  ra_deg real,
  dec_dms text,
  dec_deg real,
  vmag real,
  bv real,
  vi real,
  parallax real,
  pmra real,
  pmdec real,
  sptype text
);
sqlite> .quit
```

6.4 Adding data to table using Python script

Make a Python script to add data of stars in Hipparchos catalogue to the table “hip”.

Python Code 11: ai202209_s06_10.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 19:42:18 (CST) daisuke>
#

# importing sqlite module
import sqlite3

# database file name
```

```
file_db = 'hip.db'

# catalogue file name
file_catalogue = 'hip_main.dat'

# connecting to database
conn   = sqlite3.connect (file_db)
cursor = conn.cursor ()

# opening catalogue file
with open (file_catalogue, 'r') as fh_hip:
    # reading catalogue line-by-line
    for line in fh_hip:
        # Hipparcos Number of star
        try:
            hip = int (line[8:14])
        except:
            # printing message
            print (f'Something is wrong with following line...')
            print (f'  {line[:75]}')
            print (f'Cannot extract Hipparcos number!')
            # exit
            sys.exit (1)
        # RA in hhmmss format
        try:
            RA_hms = line[17:28].strip ()
        except:
            RA_hms = '99 99 99.99'
        # Dec in ddmmss format
        try:
            Dec_dms = line[29:40].strip ()
        except:
            Dec_dms = '-99 99 99.9'
        # V-band magnitude
        try:
            mag_V = float (line[41:46])
        except:
            mag_V = -99.99
        # RA in deg
        try:
            RA_deg = float (line[51:63])
        except:
            RA_deg = -999.99
        # Dec in deg
        try:
            Dec_deg = float (line[64:76])
        except:
            Dec_deg = -999.99
        # parallax in mas
        try:
            parallax = float (line[79:86])
        except:
            parallax = -999999.99
        # proper motion in RA
        try:
            pm_RA = float (line[87:95])
        except:
            pm_RA = -999999.99
        # proper motion in Dec
```

```
        try:
            pm_Dec = float (line[96:104])
        except:
            pm_Dec = -999999.99
        # (B-V) colour index
        try:
            colour_BV = float (line[245:251])
        except:
            colour_BV = -999.99
        # (V-I) colour index
        try:
            colour_VI = float (line[260:264])
        except:
            colour_VI = -999.99
        # spectral type
        try:
            sptype = line[435:447].strip ()
        except:
            sptype = '___NONE___'

        # SQL command to add data to table
        sql_adddata = f'insert into hip values ({hip}, ' \
            + f'"{RA_hms}", {RA_deg}, "{Dec_dms}", {Dec_deg}, ' \
            + f'{mag_V}, {colour_BV}, {colour_VI}, {parallax}, ' \
            + f'{pm_RA}, {pm_Dec}, "{sptype}");'

        # executing SQL command to add data to table
        cursor.execute (sql_adddata)

# committing transaction
conn.commit ()

# closing connection
conn.close ()
```

Execute above script to add data to the table.

```
% chmod a+x ai202209_s06_10.py
% ./ai202209_s06_10.py
% ls -lF hip*
-rw-r--r--  1 daisuke  taiwan  14270464 Oct 23 19:43 hip.db
-rw-r--r--  1 daisuke  taiwan  53316318 Oct 23 17:56 hip_main.dat
-rw-r--r--  1 daisuke  taiwan     69019 Oct 23 18:01 hip_main.readme
-rw-r--r--  1 daisuke  taiwan  29271252 Oct 23 19:38 hip_main.txt
```

Use SQLite command-line program to check the database file.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open hip.db
sqlite> .headers on
sqlite> .mode table
sqlite> select hip, ra_hms, dec_dms, vmag, bv, parallax, sptype from hip
   ...> where hip <= 10;
+-----+------------+------------+-------+--------+----------+-------------+
```

```
| hip |    ra_hms     |   dec_dms    | vmag  |   bv   | parallax |    sptype     |
+-----+-------------+-------------+-------+--------+----------+--------------+
| 1   | 00 00 00.22 | +01 05 20.4 | 9.1   | 0.482  | 3.54     | F5            |
| 2   | 00 00 00.91 | -19 29 55.8 | 9.27  | 0.999  | 21.9     | K3V           |
| 3   | 00 00 01.20 | +38 51 33.4 | 6.61  | -0.019 | 2.81     | B9            |
| 4   | 00 00 02.01 | -51 53 36.8 | 8.06  | 0.37   | 7.75     | F0V           |
| 5   | 00 00 02.39 | -40 35 28.4 | 8.55  | 0.902  | 2.87     | G8III         |
| 6   | 00 00 04.35 | +03 56 47.4 | 12.31 | 1.336  | 18.8     | M0V:          |
| 7   | 00 00 05.41 | +20 02 11.8 | 9.64  | 0.74   | 17.74    | G0            |
| 8   | 00 00 06.55 | +25 53 11.3 | 9.05  | 1.102  | 5.17     | M6e-M8.5e Tc |
| 9   | 00 00 08.48 | +36 35 09.4 | 8.59  | 1.067  | 4.81     | G5            |
| 10  | 00 00 08.70 | -50 52 01.5 | 8.59  | 0.489  | 10.76    | F6V           |
+-----+-------------+-------------+-------+--------+----------+--------------+
sqlite> select hip, ra_hms, dec_dms, vmag, bv, parallax, sptype from hip
   ...> where parallax > 300 order by parallax desc;
+--------+-------------+-------------+-------+-------+----------+--------+
|  hip   |   ra_hms    |   dec_dms   | vmag  |  bv   | parallax | sptype |
+--------+-------------+-------------+-------+-------+----------+--------+
| 70890  | 14 29 47.75 | -62 40 52.9 | 11.01 | 1.807 | 772.33   | M5Ve   |
| 71681  | 14 39 39.39 | -60 50 22.1 | 1.35  | 0.9   | 742.12   | K1V    |
| 71683  | 14 39 40.90 | -60 50 06.5 | -0.01 | 0.71  | 742.12   | G2V    |
| 87937  | 17 57 48.97 | +04 40 05.8 | 9.54  | 1.57  | 549.01   | sdM4   |
| 54035  | 11 03 20.61 | +35 58 53.3 | 7.49  | 1.502 | 392.4    | M2V    |
| 32349  | 06 45 09.25 | -16 42 47.3 | -1.44 | 0.009 | 379.21   | A0m... |
| 92403  | 18 49 48.96 | -23 50 08.8 | 10.37 | 1.51  | 336.48   | M3.5Ve |
| 16537  | 03 32 56.42 | -09 27 29.9 | 3.72  | 0.881 | 310.75   | K2V    |
| 114046 | 23 05 47.17 | -35 51 22.7 | 7.35  | 1.483 | 303.9    | M2/M3V |
+--------+-------------+-------------+-------+-------+----------+--------+
sqlite> .quit
```

6.5 Trying a SQL query using Python script

Make a Python script to carry out a SQL query. Here is an example.

Python Code 12: ai202209_s06_11.py

```python
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 19:28:12 (CST) daisuke>
#

# importing sqlite module
import sqlite3

# database file name
file_db = 'hip.db'

# connecting to database
conn   = sqlite3.connect (file_db)
cursor = conn.cursor ()

# SQL command for a query
sql_query = 'select hip, ra_hms, dec_dms, vmag, bv, parallax, sptype ' \
    + f'from hip where parallax > 200 order by parallax desc;'

# executing a SQL query
cursor.execute (sql_query)
```

```
# fetching results of query
results = cursor.fetchall ()

# printing results of query
print (f'# HIP   RA            Dec           Vmag   B-V      p      sptype')
for result in results:
    print (f'{result[0]:06d}  {result[1]}  {result[2]}  {result[3]:5.2f}', \
           f' {result[4]:7.2f}  {result[5]:5.1f}  {result[6]}')

# committing transaction
conn.commit ()

# closing connection
conn.close ()
```

Execute above script.

```
% chmod a+x ai202209_s06_11.py
% ./ai202209_s06_11.py
# HIP   RA           Dec          Vmag   B-V      p      sptype
070890  14 29 47.75  -62 40 52.9  11.01     1.81  772.3  M5Ve
071681  14 39 39.39  -60 50 22.1   1.35     0.90  742.1  K1V
071683  14 39 40.90  -60 50 06.5  -0.01     0.71  742.1  G2V
087937  17 57 48.97  +04 40 05.8   9.54     1.57  549.0  sdM4
054035  11 03 20.61  +35 58 53.3   7.49     1.50  392.4  M2V
032349  06 45 09.25  -16 42 47.3  -1.44     0.01  379.2  A0m...
092403  18 49 48.96  -23 50 08.8  10.37     1.51  336.5  M3.5Ve
016537  03 32 56.42  -09 27 29.9   3.72     0.88  310.8  K2V
114046  23 05 47.17  -35 51 22.7   7.35     1.48  303.9  M2/M3V
057548  11 47 44.04  +00 48 27.1  11.12     1.75  299.6  M4.5V
104214  21 06 50.84  +38 44 29.4   5.20     1.07  287.1  K5V
037279  07 39 18.54  +05 13 39.0   0.40     0.43  285.9  F5IV-V
104217  21 06 52.19  +38 44 03.9   6.05     1.31  285.4  K7V
091772  18 42 48.51  +59 37 20.5   9.70     1.56  284.5  K5
091768  18 42 48.22  +59 37 33.7   8.94     1.50  280.3  K5
001475  00 18 20.54  +44 01 19.0   8.09     1.56  280.3  M1V
108870  22 03 17.44  -56 46 47.3   4.69     1.06  275.8  K5V
008102  01 44 05.13  -15 56 22.4   3.49     0.73  274.2  G8V
005643  01 12 29.90  -17 00 01.9  12.10     1.85  269.1  M5.5Ve
036208  07 27 24.16  +05 14 05.2   9.84     1.57  263.3  M5
024186  05 11 35.21  -45 00 16.2   8.86     1.54  255.3  M0V
105090  21 17 17.71  -38 51 52.5   6.69     1.40  253.4  M1/M2V
110893  22 28 00.42  +57 41 49.3   9.59     1.61  249.5  M2V
030920  06 29 23.00  -02 48 44.9  11.12     1.69  242.9  M4.5Ve
072511  14 49 33.51  -26 06 21.7  11.72     1.48  235.2  M
080824  16 30 18.11  -12 39 35.0  10.10     1.60  234.5  M4
000439  00 05 20.29  -37 21 06.1   8.56     1.46  229.3  M2V
015689  03 22 05.57  -13 16 41.2  12.16  -999.99  227.4
003829  00 49 09.18  +05 23 42.7  12.37     0.55  226.9  DG
072509  14 49 32.69  -26 06 40.2  12.07     1.52  221.8  M
086162  17 36 26.41  +68 20 32.0   9.15     1.50  220.8  M3.5Vvar
085523  17 28 39.46  -46 53 35.0   9.38     1.55  220.4  K5
114110  23 06 38.89  -14 52 20.6  12.24  -999.99  216.5
057367  11 45 39.26  -64 50 26.4  11.50     0.20  216.4  DC:
113020  22 53 16.16  -14 15 43.4  10.16     1.60  212.7  M5
054211  11 05 32.13  +43 31 28.1   8.82     1.49  206.9  M2Vvar
049908  10 11 23.36  +49 27 19.7   6.60     1.33  205.2  K8V
082725  16 54 32.15  -62 24 13.5  11.72  -999.99  203.0
085605  17 29 36.19  +24 39 11.6  11.39     1.10  202.7
```

```
106440  21 33 34.02  -49 00 25.3   8.66      1.52  202.5  M1V
```

6.6 Trying one more SQL query

Find nearby B-type stars. Here is a sample Python script.

Python Code 13: ai202209_s06_12.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 19:57:15 (CST) daisuke>
#

# importing sqlite module
import sqlite3

# database file name
file_db = 'hip.db'

# connecting to database
conn   = sqlite3.connect (file_db)
cursor = conn.cursor ()

# SQL command for a query
sql_query = 'select hip, ra_hms, dec_dms, vmag, bv, parallax, sptype ' \
    + f'from hip where (parallax > 20 and sptype like "B%") ' \
    + f'order by parallax desc;'

# executing a SQL query
cursor.execute (sql_query)

# fetching results of query
results = cursor.fetchall ()

# printing results of query
print (f'# HIP   RA           Dec          Vmag   B-V      p      sptype')
for result in results:
    print (f'{result[0]:06d}  {result[1]}  {result[2]}  {result[3]:5.2f}', \
           f' {result[4]:7.2f}  {result[5]:5.1f}  {result[6]}')

# committing transaction
conn.commit ()

# closing connection
conn.close ()
```

Execute above script.

```
% chmod a+x ai202209_s06_12.py
% ./ai202209_s06_12.py
# HIP   RA           Dec          Vmag   B-V      p      sptype
030362  06 23 09.17  +08 54 26.1   9.73    -0.06   48.1  B8
049669  10 08 22.46  +11 58 01.9   1.36    -0.09   42.1  B7V
060965  12 29 51.98  -16 30 54.3   2.94    -0.01   37.1  B9.5V
014576  03 08 10.13  +40 57 20.3   2.09    -0.00   35.1  B8V
000677  00 08 23.17  +29 05 27.0   2.07    -0.04   33.6  B9p
067301  13 47 32.55  +49 18 47.9   1.85    -0.10   32.4  B3V SB
109268  22 08 13.88  -46 57 38.2   1.73    -0.07   32.2  B7IV
```

```
093805  19 06 14.95  -04 52 56.4   3.43    -0.10   26.1  B9Vn
045336  09 14 21.79  +02 18 54.1   3.89    -0.06   25.3  B9.5V
025428  05 26 17.50  +28 36 28.3   1.65    -0.13   24.9  B7III
113963  23 04 45.62  +15 12 19.3   2.49    -0.00   23.4  B9.5III
002484  00 31 32.56  -62 57 29.1   4.36    -0.06   23.4  B9V
023287  05 00 33.93  +03 36 56.9   6.65    -0.05   23.3  B9Vn
007588  01 37 42.75  -57 14 12.0   0.45    -0.16   22.7  B3Vp
090185  18 24 10.35  -34 23 03.5   1.79    -0.03   22.6  B9.5III
012394  02 39 35.22  -68 16 01.0   4.12    -0.06   21.3  B9III
116971  23 42 43.28  -14 32 41.1   4.49    -0.03   21.2  B9V
010602  02 16 30.50  -51 30 43.6   3.56    -0.12   21.1  B8IV-V
013209  02 49 58.99  +27 15 38.8   3.61    -0.10   20.4  B8Vn
074785  15 17 00.47  -09 22 58.3   2.61    -0.07   20.4  B8V
```

Try following practice.

Practice 06-07

Make a Python script to carry out a SQL query for the table "**hip**".

7 Making asteroid orbit database

Make an asteroid orbit database.

7.1 Downloading asteroid orbit catalogue

Make a Python script to download asteroid orbit catalogue from Minor Planet Center. Here is an example.

Python Code 14: ai202209_s06_13.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 20:06:48 (CST) daisuke>
#

# importing urllib module
import urllib.request

# importing ssl module
import ssl

# allow insecure downloading
ssl._create_default_https_context = ssl._create_unverified_context

# URL of data file
url_data = 'https://www.minorplanetcenter.net/iau/MPCORB/MPCORB.DAT.gz'

# output file name
file_output = 'mpcorb.dat.gz'

# printing status
print (f'Now, fetching {url_data}...')

# opening URL
with urllib.request.urlopen (url_data) as fh_read:
    # reading data
    data_byte = fh_read.read ()
```

```
# printing status
print (f'Finished fetching {url_data}!')

# printing status
print (f'Now, writing the data into file "{file_output}"...')

# opening file for writing
with open (file_output, 'wb') as fh_write:
    # writing data
    fh_write.write (data_byte)

# printing status
print (f'Finished writing the data into file "{file_output}"!')
```

Execute above script to download the file.

```
% chmod a+x ai202209_s06_13.py
% ./ai202209_s06_13.py
Now, fetching https://www.minorplanetcenter.net/iau/MPCORB/MPCORB.DAT.gz...
Finished fetching https://www.minorplanetcenter.net/iau/MPCORB/MPCORB.DAT.gz!
Now, writing the data into file "mpcorb.dat.gz"...
Finished writing the data into file "mpcorb.dat.gz"!
% ls -lF mpcorb.dat.gz
-rw-r--r--  1 daisuke  taiwan  74080809 Oct 23 20:08 mpcorb.dat.gz
```

7.2 Reading asteroid orbit catalogue

Make a Python script to open and read the asteroid orbit catalogue. Here is an example.

Python Code 15: ai202209_s06_14.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 20:33:49 (CST) daisuke>
#

# importing gzip module
import gzip

# catalogue file name
file_catalogue = 'mpcorb.dat.gz'

# opening catalogue file
with gzip.open (file_catalogue, 'rb') as fh:
    # reading catalogue line-by-line
    for line in fh:
        # number of provisional designation
        try:
            designation = line[0:7].strip ().decode ('utf-8')
        except:
            continue
        # absolute magnitude
        try:
            absmag = float (line[8:13])
        except:
            absmag = -999.99
        # mean anomaly
        try:
```

```
            M = float (line[26:35])
        except:
            M = -999.99
        # argument of perihelion
        try:
            peri = float (line[37:46])
        except:
            peri = -999.99
        # longitude of ascending node
        try:
            node = float (line[48:57])
        except:
            node = -999.99
        # inclination
        try:
            i = float (line[59:68])
        except:
            i = -999.99
        # eccentricity
        try:
            e = float (line[70:79])
        except:
            e = -999.99
        # semimajor axis
        try:
            a = float (line[92:103])
        except:
            a = -999.99
        # number of observations
        try:
            nobs = int (line[117:122])
        except:
            nobs = -999
        # residual
        try:
            residual = float (line[137:141])
        except:
            residual = -999.99
        # 4-hexdigit flags
        try:
            flag = line[161:165].strip ().decode ('utf-8')
        except:
            flag = '9999'
        # readable name
        try:
            name = line[166:194].strip ().decode ('utf-8')
        except:
            name = '__NONE__'
        # last observation date
        try:
            lastobs = int (line[194:202])
        except:
            lastobs = 99999999

        # skip when reading the header
        if ( (a < -999.0) and (e < -999.0) and (i < -999.0) \
             and (peri < -999.0) and (node < -999.0) and (M < -999.0) ):
            continue
```

```
        # printing extracted data
        print (f'designation = {designation}')
        print (f'  name      = {name}')
        print (f'  absmag    = {absmag}')
        print (f'  M         = {M}')
        print (f'  peri      = {peri}')
        print (f'  node      = {node}')
        print (f'  i         = {i}')
        print (f'  e         = {e}')
        print (f'  a         = {a}')
        print (f'  nobs      = {nobs}')
        print (f'  residual  = {residual}')
        print (f'  flag      = {flag}')
        print (f'  lastobs   = {lastobs}')
```

Execute above script.

```
% chmod a+x ai202209_s06_14.py
% ./ai202209_s06_14.py > mpcorb.txt
% ls -lF mpcorb.*
-rw-r--r--  1 daisuke  taiwan   74080809 Oct 23 20:08 mpcorb.dat.gz
-rw-r--r--  1 daisuke  taiwan  344353066 Oct 23 20:46 mpcorb.txt
% head -20 mpcorb.txt
designation = 00001
  name      = (1) Ceres
  absmag    = 3.32
  M         = 334.32723
  peri      = 73.53158
  node      = 80.26642
  i         = 10.5868
  e         = 0.0786358
  a         = 2.7666192
  nobs      = 7259
  residual  = 0.65
  flag      = 0000
  lastobs   = 20220916
designation = 00002
  name      = (2) Pallas
  absmag    = 4.12
  M         = 315.09111
  peri      = 310.84262
  node      = 172.91791
  i         = 34.92715
```

7.3 Constructing asteroid orbit database

Make a Python script to construct asteroid orbit database. Here is an example.

Python Code 16: ai202209_s06_15.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 21:03:35 (CST) daisuke>
#

# importing gzip module
import gzip
```

```
# importing sqlite module
import sqlite3

# catalogue file name
file_catalogue = 'mpcorb.dat.gz'

# database file name
file_db = 'mpcorb.db'

# connecting to database
conn   = sqlite3.connect (file_db)
cursor = conn.cursor ()

# SQL command for making a table
sql_maketable = f'create table mpcorb (designation text primary key, ' \
    + f'name text, a real, e real, i real, node real, peri real, M real, ' \
    + f'nobs integer, residual real, flag text, lastobs integer, ' \
    + f'absmag real);'

# making a table
cursor.execute (sql_maketable)

# opening catalogue file
with gzip.open (file_catalogue, 'rb') as fh:
    # reading catalogue line-by-line
    for line in fh:
        # number of provisional designation
        try:
            designation = line[0:7].strip ().decode ('utf-8')
        except:
            continue
        # absolute magnitude
        try:
            absmag = float (line[8:13])
        except:
            absmag = -999.99
        # mean anomaly
        try:
            M = float (line[26:35])
        except:
            M = -999.99
        # argument of perihelion
        try:
            peri = float (line[37:46])
        except:
            peri = -999.99
        # longitude of ascending node
        try:
            node = float (line[48:57])
        except:
            node = -999.99
        # inclination
        try:
            i = float (line[59:68])
        except:
            i = -999.99
        # eccentricity
        try:
            e = float (line[70:79])
```

```
        except:
            e = -999.99
        # semimajor axis
        try:
            a = float (line[92:103])
        except:
            a = -999.99
        # number of observations
        try:
            nobs = int (line[117:122])
        except:
            nobs = -999
        # residual
        try:
            residual = float (line[137:141])
        except:
            residual = -999.99
        # 4-hexdigit flags
        try:
            flag = line[161:165].strip ().decode ('utf-8')
        except:
            flag = '9999'
        # readable name
        try:
            name = line[166:194].strip ().decode ('utf-8')
        except:
            name = '__NONE__'
        # last observation date
        try:
            lastobs = int (line[194:202])
        except:
            lastobs = 99999999

        # skip when reading the header
        if ( (a < -999.0) and (e < -999.0) and (i < -999.0) \
             and (peri < -999.0) and (node < -999.0) and (M < -999.0) ):
            continue

        # SQL command to add data to table
        sql_adddata = f'insert into mpcorb values ("{designation}", ' \
            + f'"{name}", {a}, {e}, {i}, {node}, {peri}, {M}, ' \
            + f'{nobs}, {residual}, "{flag}", {lastobs}, {absmag});'

        # adding data to table
        cursor.execute (sql_adddata)

# committing transaction
conn.commit ()

# closing connection
conn.close ()
```

Execute above script to make a table and add data to the table.

```
% chmod a+x ai202209_s06_15.py
% ./ai202209_s06_15.py
% ls -lF mpcorb.*
-rw-r--r--  1 daisuke  taiwan   74080809 Oct 23 20:08 mpcorb.dat.gz
-rw-r--r--  1 daisuke  taiwan  164065280 Oct 23 21:04 mpcorb.db
```

```
-rw-r--r--  1 daisuke  taiwan  344353066 Oct 23 20:46 mpcorb.txt
% file mpcorb.db
mpcorb.db: SQLite 3.x database, last written using SQLite version 3026000, file
counter 2, database pages 40055, cookie 0x1, schema 4, UTF-8, version-valid-for
2
```

Use SQLite command-line program to check the database file.

```
% sqlite3
SQLite version 3.39.4 2022-09-29 15:55:41
Enter ".help" for usage hints.
Connected to a transient in-memory database.
Use ".open FILENAME" to reopen on a persistent database.
sqlite> .open mpcorb.db
sqlite> .tables
mpcorb
sqlite> .schema --indent mpcorb
CREATE TABLE mpcorb(
  designation text primary key,
  name text,
  a real,
  e real,
  i real,
  node real,
  peri real,
  M real,
  nobs integer,
  residual real,
  flag text,
  lastobs integer,
  absmag real
);
sqlite> .headers on
sqlite> .mode table
sqlite> select name, a, e, i, absmag from mpcorb
   ...> where (absmag < 4.0 and absmag > 0.0) order by absmag;
+---------------------------+-------------+-----------+----------+--------+
|           name            |      a      |     e     |    i     | absmag |
+---------------------------+-------------+-----------+----------+--------+
| (136108) Haumea           | 42.941274   | 0.1997438 | 28.2115  | 0.23   |
| (90377) Sedna             | 521.2989572 | 0.8534734 | 11.9309  | 1.54   |
| (225088) Gonggong         | 67.3689423  | 0.4976969 | 30.61495 | 1.86   |
| (90482) Orcus             | 39.0973385  | 0.2292832 | 20.57341 | 2.19   |
| (50000) Quaoar            | 43.471578   | 0.0409873 | 7.99122  | 2.42   |
| (532037) 2013 FY27        | 58.5354385  | 0.3998662 | 33.28776 | 3.15   |
| (4) Vesta                 | 2.3619872   | 0.0884019 | 7.14078  | 3.2    |
| (1) Ceres                 | 2.7666192   | 0.0786358 | 10.5868  | 3.32   |
| (174567) Varda            | 45.8587843  | 0.1455804 | 21.52094 | 3.46   |
| (28978) Ixion             | 39.6649849  | 0.2471526 | 19.64351 | 3.47   |
| (55565) 2002 AW197        | 46.9062914  | 0.1263019 | 24.42113 | 3.47   |
| 2014 UZ224                | 108.9254846 | 0.644317  | 26.78842 | 3.48   |
| (229762) G!kun||'homdima  | 73.6679589  | 0.4892783 | 23.38139 | 3.5    |
| (55636) 2002 TX300        | 43.555102   | 0.1254113 | 25.83133 | 3.53   |
| 2021 DR15                 | 67.2246997  | 0.4369596 | 30.67996 | 3.61   |
| (307261) 2002 MS4         | 41.908802   | 0.1440324 | 17.70697 | 3.62   |
| (145452) 2005 RN43        | 41.866602   | 0.0305685 | 19.21627 | 3.7    |
| (208996) 2003 AZ84        | 39.3162234  | 0.1801605 | 13.56256 | 3.77   |
| (20000) Varuna            | 42.8087201  | 0.0577897 | 17.18605 | 3.79   |
```

```
| (303775) 2005 QU182       | 114.6265834 | 0.6767155 | 14.02307 | 3.79   |
| (55637) 2002 UX25         | 42.8742952  | 0.1417437 | 19.38789 | 3.86   |
| (589683) 2010 RF43        | 49.713736   | 0.2422064 | 30.55977 | 3.87   |
| (202421) 2005 UQ513       | 43.5918468  | 0.1408421 | 25.70339 | 3.92   |
| (523692) 2014 EZ51        | 52.1166152  | 0.2314198 | 10.30132 | 3.92   |
| (84522) 2002 TC302        | 55.5415404  | 0.2941881 | 34.95848 | 3.93   |
| (574372) 2010 JO179       | 77.8715398  | 0.4946613 | 32.02267 | 3.93   |
| 2018 VG18                 | 81.8549833  | 0.5297237 | 24.22901 | 3.94   |
+---------------------------+-------------+-----------+----------+--------+
sqlite> .quit
```

7.4 Trying some SQL queries

Make a Python script to carry out SQL queries.

Python Code 17: ai202209_s06_16.py

```python
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 21:24:47 (CST) daisuke>
#

# importing sqlite module
import sqlite3

# database file name
file_db = 'mpcorb.db'

# connecting to database
conn   = sqlite3.connect (file_db)
cursor = conn.cursor ()

# SQL command for a query
sql_query = 'select name, a, e, i, node, peri, M, nobs, residual, ' \
    + f'flag, lastobs, absmag from mpcorb ' \
    + f'where (a >= 1000.0) order by a desc;'

# executing a SQL query
cursor.execute (sql_query)

# fetching results of query
results = cursor.fetchall ()

# printing results of query
print (f'# name, a, e, i, node, peri, M, absmag')
for result in results:
    print (f'{result[0]:24s}  {result[1]:8.3f}  {result[2]:5.3f} ', \
           f'{result[3]:6.2f} {result[4]:6.2f} {result[5]:6.2f} ', \
           f'{result[6]:6.2f}  {result[11]:7.2f}')

# committing transaction
conn.commit ()

# closing connection
conn.close ()
```

Execute above script.

```
% chmod a+x ai202209_s06_16.py
% ./ai202209_s06_16.py
# name, a, e, i, node, peri, M, absmag
2010 LN135                  3640.394  1.000   64.73 184.70 181.42     0.02     14.08
2017 MB7                    3549.257  0.999   55.71  58.26  80.46     0.00     14.20
2014 FE72                   1608.337  0.978   20.70 336.97 133.42     0.32      6.19
(541132) Leleakuhonua       1355.189  0.952   11.66 300.87 117.60   359.60      5.57
2021 RR205                  1265.746  0.956    7.64 108.29 209.05     0.24      6.74
2022 QE78                   1241.531  0.996   36.55 119.94   0.20   359.97      9.36
(308933) 2006 SQ372         1115.455  0.978   19.43 197.37 122.71     0.15      7.94
2012 DR30                   1050.541  0.986   78.00 341.56 195.26     0.12      7.12
2013 BL76                   1029.360  0.992   98.57 180.01 166.05     0.11     10.88
```

Try one more SQL query.

Python Code 18: ai202209_s06_17.py

```
#!/usr/pkg/bin/python3.9

#
# Time-stamp: <2022/10/23 22:28:59 (CST) daisuke>
#

# importing sqlite module
import sqlite3

# database file name
file_db = 'mpcorb.db'

# connecting to database
conn   = sqlite3.connect (file_db)
cursor = conn.cursor ()

# SQL command for a query
sql_query = 'select name, a, e, i, node, peri, M, nobs, residual, ' \
    + f'flag, lastobs, absmag from mpcorb ' \
    + f'where (i >= 170.0) order by i desc;'

# executing a SQL query
cursor.execute (sql_query)

# fetching results of query
results = cursor.fetchall ()

# printing results of query
print (f'# name, a, e, i, node, peri, M, absmag')
for result in results:
    print (f'{result[0]:24s}  {result[1]:8.3f}  {result[2]:5.3f} ', \
           f'{result[3]:6.2f} {result[4]:6.2f} {result[5]:6.2f} ', \
           f'{result[6]:6.2f}  {result[11]:7.2f}')

# committing transaction
conn.commit ()

# closing connection
conn.close ()
```

Execute above script.

```
% chmod a+x ai202209_s06_17.py
% ./ai202209_s06_17.py
# name, a, e, i, node, peri, M, absmag
2022 FN12                  136.255  0.566  178.46 254.65  47.25    1.24     6.34
(582301) 2015 RM306        243.329  0.953  175.98  52.78  44.58    0.39    11.07
2013 LA2                     5.683  0.467  175.09 243.90 325.29  255.48    16.94
(434620) 2005 VD             6.673  0.252  172.87 173.37 178.26  343.44    14.30
2022 FM12                  158.770  0.663  172.45  13.42 175.57  359.36     6.54
2006 LM1                    37.184  0.900  172.14 120.69 201.94  359.55    14.80
2021 XZ3                    14.199  0.781  172.09 350.59 331.50  358.76    14.82
2016 EJ203                  65.600  0.959  170.96   7.08 227.36    4.13    18.10
2018 TL6                     8.292  0.792  170.93  45.60  78.49   54.43    19.90
2014 UV114                  13.000  0.690  170.88  53.31   7.30   61.76    15.80
2014 CW14                   32.709  0.868  170.75 181.79  80.22   14.84    14.21
(330759) 2008 SO218          8.125  0.562  170.35 348.45 354.62  196.36    12.90
2021 YP                      6.146  0.687  170.20 175.22 156.89    7.44    17.57
```

Try following practice.

Practice 06-08

Make a Python script to carry out a SQL query for the table "`mpcorb`".

8 Practice A: Exoplanet database

Visit following web page. (Fig. 21)

- NASA Exoplanet Archive: `https://exoplanetarchive.ipac.caltech.edu/`

Make an exoplanet database.

1. Move the mouse cursor to the menu "Data". (Fig. 22)
2. Click the menu "Planetary Systems" and go to the planetary system table. (Fig. 23)
3. Move the mouse cursor to the menu "Download Table". (Fig. 24)
4. Choose "Download All Columns" and "Download All Rows".
5. Click the button "Download Table" at the bottom of the pull-down menu "Download Table".
6. Check the CSV file you have downloaded.
7. Design your own table for exoplanet database.
8. Make a Python script to create a table.
9. Make a Python script to construct an exoplanet database.
10. Make a Python script to carry out SQL query for the exoplanet database.
11. Show the result of your query.

9 Practice B: Variable star database

Visit following web page. (Fig. 25)

- General Catalogue of Variable Stars new version (GCVS 5.1): `http://www.sai.msu.su/gcvs/gcvs/gcvs5/htm/`

1. Download GCVS 5.1 (file name "`gcvs5.txt`"). (Fig. 26)

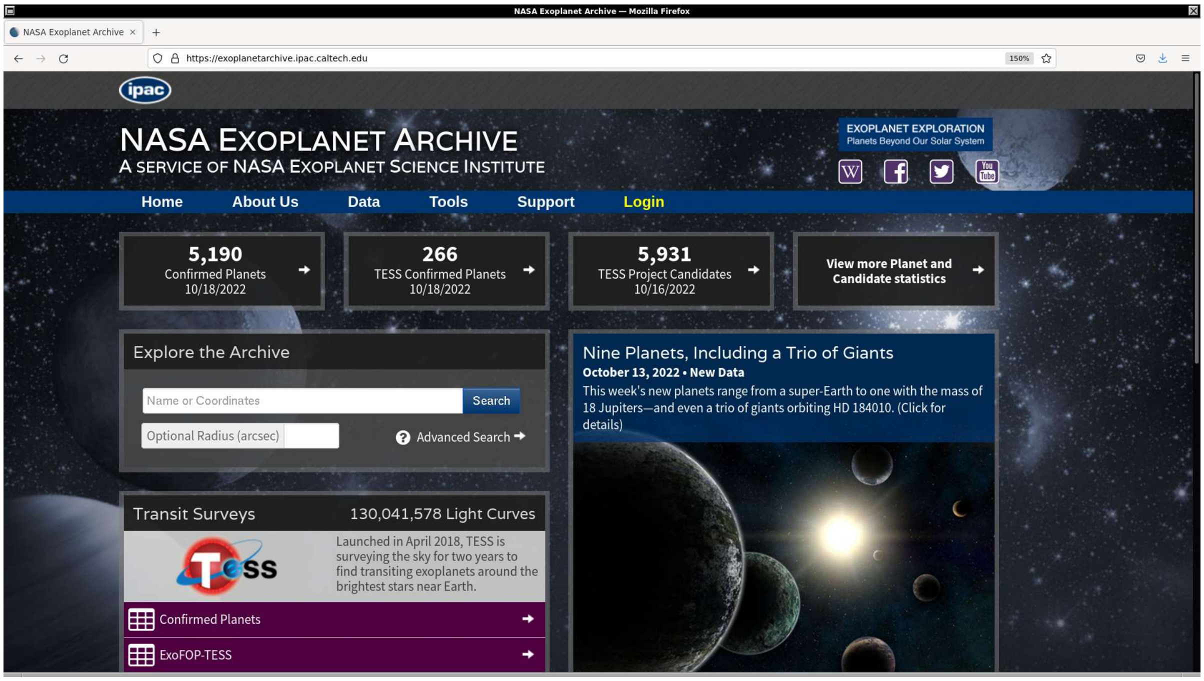


Figure 21: The NASA Exoplanet Archive website.

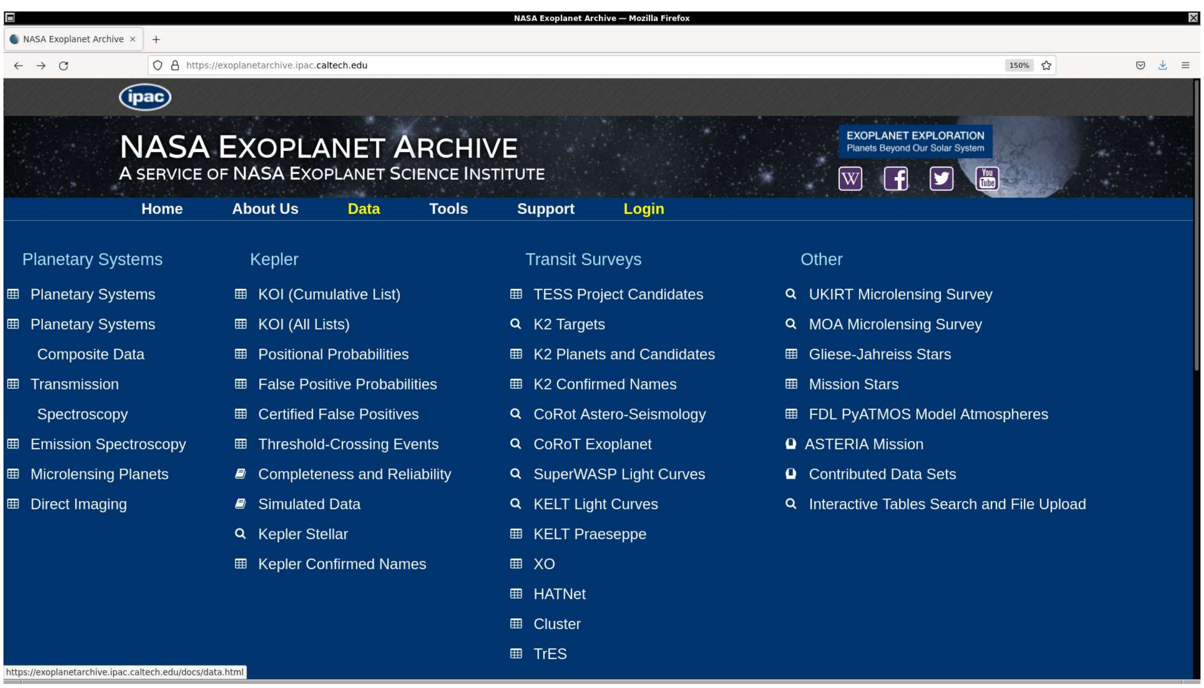


Figure 22: The menu "Data" of the NASA Exoplanet Archive website.

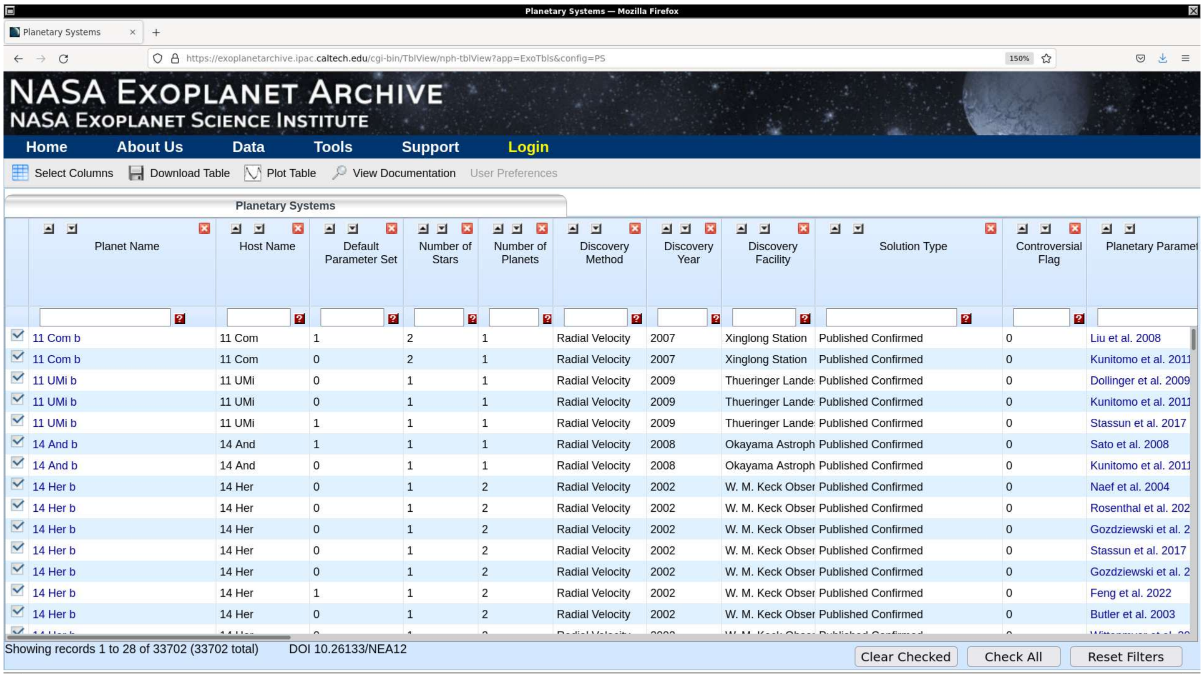


Figure 23: The planetary system table of the NASA Exoplanet Archive website.

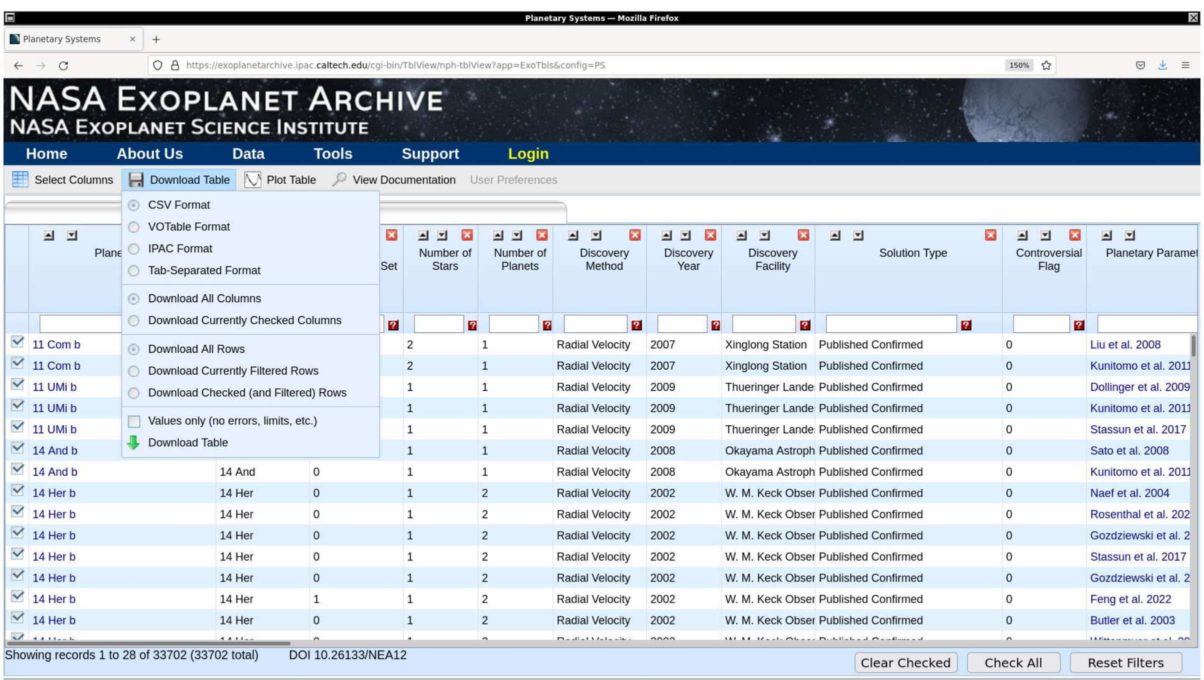


Figure 24: The "Download Table" menu of the NASA Exoplanet Archive website.

2. Design your own table for variable star database.
3. Make a Python script to create a table.
4. Make a Python script to construct a variable star database.
5. Make a Python script to carry out SQL query for the variable star database.
6. Show the result of your query.

Figure 25: The GCVS (General Catalogue of Variable Stars) website.

10 Practice C: Brown dwarf database

Visit following web page.

- List of Brown Dwarfs: `http://www.johnstonsarchive.net/astro/browndwarflist.html`

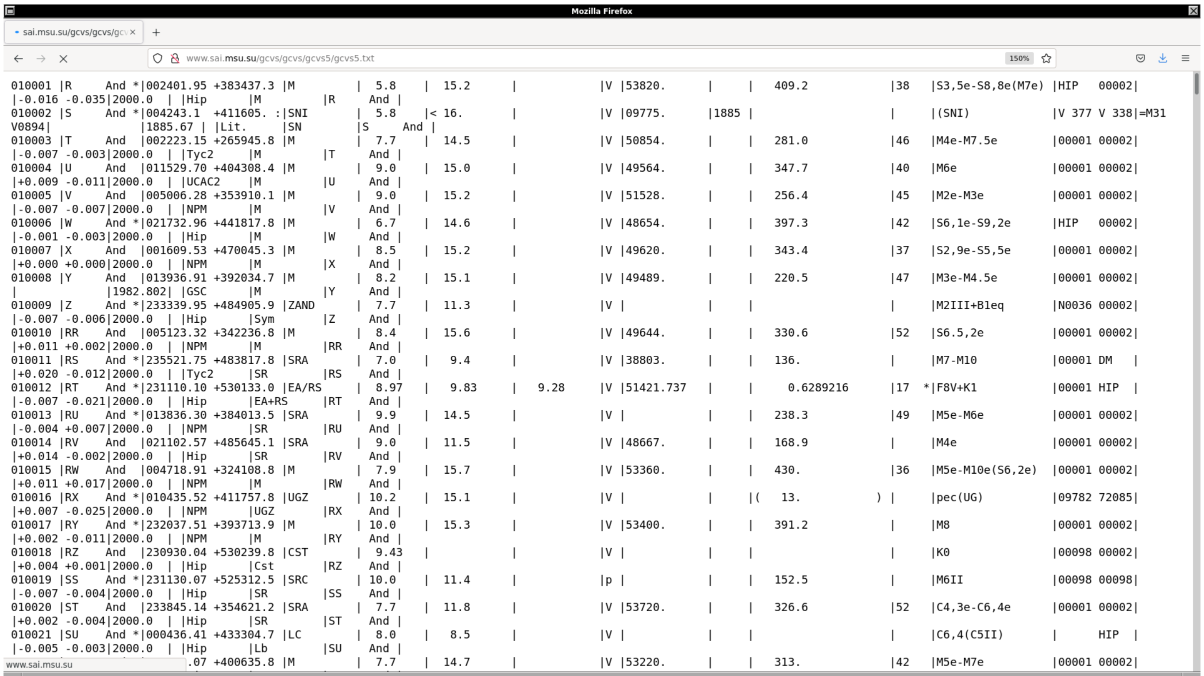
Make a brown dwarf database.

1. Design your own table for brown dwarf database.
2. Make a Python script to create a table.
3. Make a Python script to construct a brown dwarf database.
4. Make a Python script to carry out SQL query for the brown dwarf database.
5. Show the result of your query.

11 For your further reading

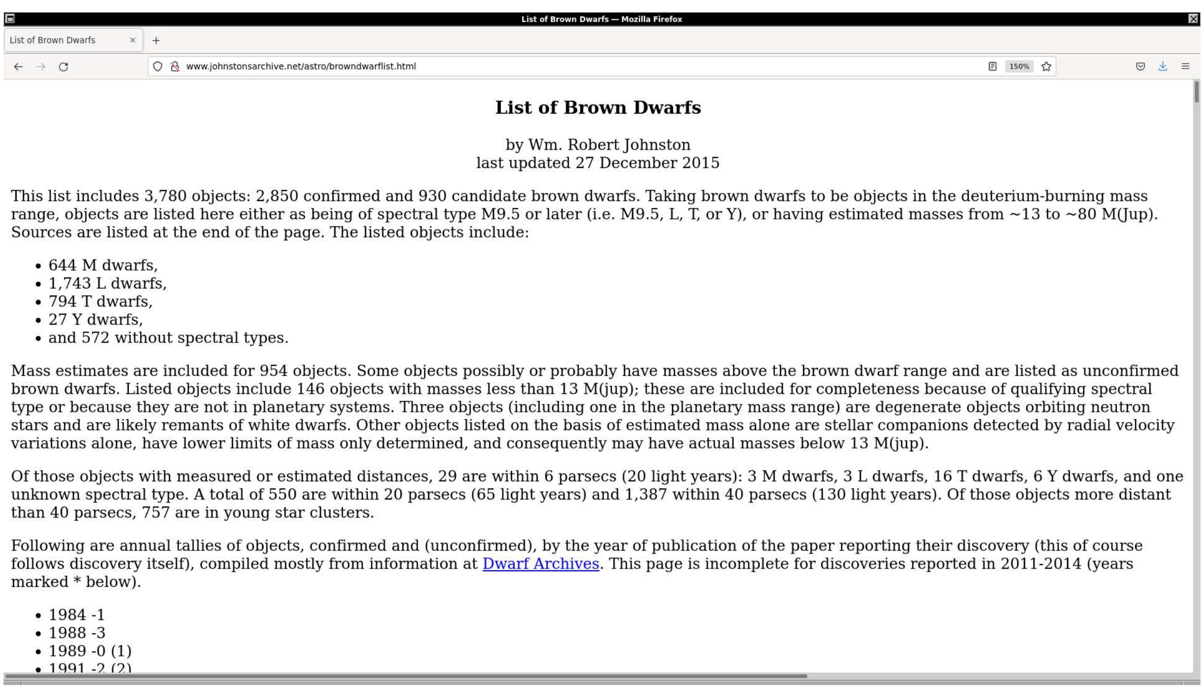
Read following document to learn more about "`sqlite3`" module of Python.

- sqlite3: `https://docs.python.org/3/library/sqlite3.html`




```
010001 |R     And *|002401.95 +383437.3 |M         |  5.8    |  15.2      |           |V |53820.     |     |   409.2          |38   |S3,5e-S8,8e(M7e) |HIP   00002|
|-0.016 -0.035|2000.0  | |Hip       |M          |R     And |
010002 |S     And *|004243.1  +411605. :|SNI       |  5.8    |< 16.      |           |V |09775.     |1885 |                  |     |(SNI)            |V 377 V 338|=M31
V0894|             |1885.67 | |Lit.      |SN         |S     And |
010003 |T     And  |002223.15 +265945.8 |M         |  7.7    |  14.5      |           |V |50854.     |     |   281.0          |46   |M4e-M7.5e        |00001 00002|
|-0.007 -0.003|2000.0  | |Tyc2      |M          |T     And |
010004 |U     And  |011529.70 +404308.4 |M         |  9.0    |  15.0      |           |V |49564.     |     |   347.7          |40   |M6e              |00001 00002|
|+0.009 -0.011|2000.0  | |UCAC2     |M          |U     And |
010005 |V     And  |005006.28 +353910.1 |M         |  9.0    |  15.2      |           |V |51528.     |     |   256.4          |45   |M2e-M3e          |00001 00002|
|-0.007 -0.007|2000.0  | |NPM       |M          |V     And |
010006 |W     And *|021732.96 +441817.8 |M         |  6.7    |  14.6      |           |V |48654.     |     |   397.3          |42   |S6,1e-S9,2e      |HIP   00002|
|-0.001 -0.003|2000.0  | |Hip       |M          |W     And |
010007 |X     And  |001609.53 +470045.3 |M         |  8.5    |  15.2      |           |V |49620.     |     |   343.4          |37   |S2,9e-S5,5e      |00001 00002|
|+0.000 +0.000|2000.0  | |NPM       |M          |X     And |
010008 |Y     And  |013936.91 +392034.7 |M         |  8.2    |  15.1      |           |V |49489.     |     |   220.5          |47   |M3e-M4.5e        |00001 00002|
|             |1982.802| |GSC       |M          |Y     And |
010009 |Z     And *|233339.95 +484905.9 |ZAND      |  7.7    |  11.3      |           |V |           |     |                  |     |M2III+B1eq       |N0036 00002|
|-0.007 -0.006|2000.0  | |Hip       |Sym        |Z     And |
010010 |RR    And  |005123.32 +342236.8 |M         |  8.4    |  15.6      |           |V |49644.     |     |   330.6          |52   |S6.5,2e          |00001 00002|
|+0.011 +0.002|2000.0  | |NPM       |M          |RR    And |
010011 |RS    And *|235521.75 +483817.8 |SRA       |  7.0    |   9.4      |           |V |38803.     |     |   136.           |     |M7-M10           |00001 DM   |
|+0.020 -0.012|2000.0  | |Tyc2      |SR         |RS    And |
010012 |RT    And *|231110.10 +530133.0 |EA/RS     |  8.97   |   9.83     |   9.28    |V |51421.737  |     |     0.6289216    |17  *|F8V+K1           |00001 HIP  |
|-0.007 -0.021|2000.0  | |Hip       |EA+RS      |RT    And |
010013 |RU    And *|013836.30 +384013.5 |SRA       |  9.9    |  14.5      |           |V |           |     |   238.3          |49   |M5e-M6e          |00001 00002|
|-0.004 +0.007|2000.0  | |NPM       |SR         |RU    And |
010014 |RV    And  |021102.57 +485645.1 |SRA       |  9.0    |  11.5      |           |V |48667.     |     |   168.9          |     |M4e              |00001 00002|
|+0.014 -0.002|2000.0  | |Hip       |SR         |RV    And |
010015 |RW    And  |004718.91 +324108.8 |M         |  7.9    |  15.7      |           |V |53360.     |     |   430.           |36   |M5e-M10e(S6,2e)  |00001 00002|
|+0.011 +0.017|2000.0  | |NPM       |M          |RW    And |
010016 |RX    And *|010435.52 +411757.8 |UGZ       | 10.2    |  15.1      |           |V |           |     |(   13.         ) |     |pec(UG)          |09782 72085|
|+0.007 -0.025|2000.0  | |NPM       |UGZ        |RX    And |
010017 |RY    And *|232037.51 +393713.9 |M         | 10.0    |  15.3      |           |V |53400.     |     |   391.2          |     |M8               |00001 00002|
|+0.002 -0.011|2000.0  | |NPM       |M          |RY    And |
010018 |RZ    And  |230930.04 +530239.8 |CST       |  9.43   |            |           |V |           |     |                  |     |K0               |00098 00002|
|+0.004 +0.001|2000.0  | |Hip       |Cst        |RZ    And |
010019 |SS    And *|231130.07 +525312.5 |SRC       | 10.0    |  11.4      |           |p |           |     |   152.5          |     |M6II             |00098 00098|
|-0.007 -0.004|2000.0  | |Hip       |SR         |SS    And |
010020 |ST    And  |233845.14 +354621.2 |SRA       |  7.7    |  11.8      |           |V |53720.     |     |   326.6          |52   |C4,3e-C6,4e      |00001 00002|
|+0.002 -0.004|2000.0  | |Hip       |SR         |ST    And |
010021 |SU    And *|000436.41 +433304.7 |LC        |  8.0    |   8.5      |           |V |           |     |                  |     |C6,4(C5II)       |      HIP  |
|-0.005 -0.003|2000.0  | |Hip       |Lb         |SU    And |
                   .07 +400635.8 |M         |  7.7    |  14.7      |           |V |53220.     |     |   313.           |42   |M5e-M7e          |00001 00002|
```

Figure 26: The GCVS 5.1 data file.



List of Brown Dwarfs

by Wm. Robert Johnston
last updated 27 December 2015

This list includes 3,780 objects: 2,850 confirmed and 930 candidate brown dwarfs. Taking brown dwarfs to be objects in the deuterium-burning mass range, objects are listed here either as being of spectral type M9.5 or later (i.e. M9.5, L, T, or Y), or having estimated masses from ~13 to ~80 M(Jup). Sources are listed at the end of the page. The listed objects include:

- 644 M dwarfs,
- 1,743 L dwarfs,
- 794 T dwarfs,
- 27 Y dwarfs,
- and 572 without spectral types.

Mass estimates are included for 954 objects. Some objects possibly or probably have masses above the brown dwarf range and are listed as unconfirmed brown dwarfs. Listed objects include 146 objects with masses less than 13 M(jup); these are included for completeness because of qualifying spectral type or because they are not in planetary systems. Three objects (including one in the planetary mass range) are degenerate objects orbiting neutron stars and are likely remants of white dwarfs. Other objects listed on the basis of estimated mass alone are stellar companions detected by radial velocity variations alone, have lower limits of mass only determined, and consequently may have actual masses below 13 M(jup).

Of those objects with measured or estimated distances, 29 are within 6 parsecs (20 light years): 3 M dwarfs, 3 L dwarfs, 16 T dwarfs, 6 Y dwarfs, and one unknown spectral type. A total of 550 are within 20 parsecs (65 light years) and 1,387 within 40 parsecs (130 light years). Of those objects more distant than 40 parsecs, 757 are in young star clusters.

Following are annual tallies of objects, confirmed and (unconfirmed), by the year of publication of the paper reporting their discovery (this of course follows discovery itself), compiled mostly from information at Dwarf Archives. This page is incomplete for discoveries reported in 2011-2014 (years marked * below).

- 1984 -1
- 1988 -3
- 1989 -0 (1)
- 1991 -2 (2)

Figure 27: The "List of Brown Dwarfs" web page.

12 Assignment

1. Learn about SQL language. Summarise what you have studied.
2. Near-Earth asteroid database
   (a) Visit Minor Planet Center website.
      - `https://minorplanetcenter.net/data`

   (b) Download the file "`NEA.txt`".
   (c) Learn about the catalogue format, and design your own table for the near-Earth asteroid database.
   (d) Make a Python script to create a table for your near-Earth asteroid database.
   (e) Make a Python script to read the catalogue file.
   (f) Make a Python script to add data of near-Earth asteroids to the table.
   (g) Make a Python script to carry out a SQL query for your near-Earth asteroid database.
   (h) Execute the script and show the result of your query.
3. Quasar database
   (a) Visit the Million Quasar Catalog website.
      - `https://quasars.org/milliquas.htm`

   (b) Download the catalogue file.
   (c) Learn about the catalogue format, and design your own table for the quasar database.
   (d) Make a Python script to create a table for your quasar database.
   (e) Make a Python script to read the catalogue file.
   (f) Make a Python script to add data of quasars to the table.
   (g) Make a Python script to carry out a SQL query for your quasar database.
   (h) Execute the script and show the result of your query.
4. Supernova database
   (a) Visit the Open Supernova Catalog web page.
      - `https://github.com/astrocatalogs/supernovae`

   (b) Download the catalogue files.
   (c) Learn about the catalogue format, and design your own table for the supernova database.
   (d) Make a Python script to create a table for your supernova database.
   (e) Make a Python script to read the catalogue file.
   (f) Make a Python script to add data of supernovae to the table.
   (g) Make a Python script to carry out a SQL query for your supernova database.
   (h) Execute the script and show the result of your query.