

# SPJRUD to SQL

---

## Purpose

The purpose of this project is to build a simple library to convert Relational Algebra operations (SPJRUD) to SQL queries. The project presents itself as a simple python library to import in a project or in an interactive python console. See Usage for more information.

## Used operations (SPJRUD)

---

Below a list of the operations used in this project, with an explanation of what they are.

### Selection

The selection operation selects tuples which satisfy the given predicate from a relation.

### Projection

The projection operation projects column(s) that satisfy a given predicate.

### Join

The projection operation combines two relations into one with a cartesian product.

### Rename

The rename operation allows us to rename column(s) in the output relation.

### Union

The results of the union operation are tuples, which are present in, at least one, of the two relations.

### Difference

The result of the difference operation are tuples, which are present in the first relation but not in the second relation.

## Usage

---

Here is an example of how the project is supposed to be used in interactive mode :

**Import the module and set the database.**

A function `set_db()` is included to set a database. By default, the extension of the databases files is '.db' but a custom extension can be set as a second parameter for the function.

```
>>> from spjrud_to_sql import *
>>> set_db('database')
```

A file called "database.db" will then be created.

**Fill or create a database.**

If the database is empty, it can be filled with the given function with normal SQLite queries.

```
>>> execute('' CREATE TABLE Rouge (A TEXT, B TEXT, C NUMERIC)'' )
>>> execute("INSERT INTO Rouge VALUES ('abc', 'klm', 1)")
>>> execute("INSERT INTO Rouge VALUES ('def', 'nop', 2)")
>>> execute("INSERT INTO Rouge VALUES ('hij', 'qrs', 3)")
```

Additionally, the `create_table()` function can be used to create a table based on another table which already exists by using the provided relational algebra operations explained below.

```
>>> create_table('Magenta', S(Attr('A'), Cst('abc'), Rel('Rouge')))
Table Magenta successfully created.
[('abc', 'klm', 1)]
```

**SPJRUD operations.**

Each operations can be used within the `execute()` function by typing the first letter of the operation. The `Relation` , `Attribute` and `Constant` objects can be respectively instanciate by `Rel()` , `Attr()` and `Cst()` . So, for example, if you want to project the attribute A in the relation called 'Rouge' :

```
>>> execute(P([Attr('A')], Rel('Rouge')))
[('abc',), ('def',), ('hij',)]
```

It will automatically print the output of the query. If you want to get the last output printed, you can get it with the `Database.current.output` variable.

## Operation's arguments

Operation	Arguments
Select	An attribute, an attribute or a constant.
Project	A list of attributes, a relation/subquery.
Join	A relation/subquery, a relation/subquery
Rename	An attribute, a string, a relation/subquery
Union	A relation/subquery, a relation/subquery
Diff	A relation/subquery, a relation/subquery