WARRU

Free Database Learning Platform

Fouad DAHAK
Higher school of computer science

https://www.youtube.com/c/dahakfouads

https://www.linkedin.com/in/dahakfouad/

WARRU

Version 1.3.2.7

User Manual

By

Fouad DAHAK

ECOLE NATIONALE SUPÉRIEURE D'INFORMATIQUE

April 2020

Table of Contents

I. Presentation 4
II. Downloading and Installing Warru 4
III. Architecture 10
IV. Getting started 11
1. User Interface 11
2. New Project 11
3. Open Existing Project 14
4. Query Editor 15
4.1. Query Editor 15
4.2. Query Parsing 16
4.3. Saving and exporting queries 16
V. Algebraic Language 17
1. Algebraic Operation Template 17
2. Algebraic Operation Syntax 18
2.1. Operations 18
2.2. Conditions 21
2.3. Aliases 22
2.4. Comments 22
VI. SQL Queries 23
VII. Samples 24
1. DZFoot 24
2. Sakila 24
3. World 24

I. Presentation

Warru is an Algebraic Editor built on SQLite allowing to make queries with relational algebra and SQL. Algebraic operations are all implemented in Warru. The Warru algebraic interpreter also allows you to see the execution time of a query and the memory space consumed in addition to the construction of its corresponding algebraic tree.

Warru is an educational tool with an integrated Help allowing to have quickly the syntax of any algebraic operation.

II. Downloading and Installing Warru

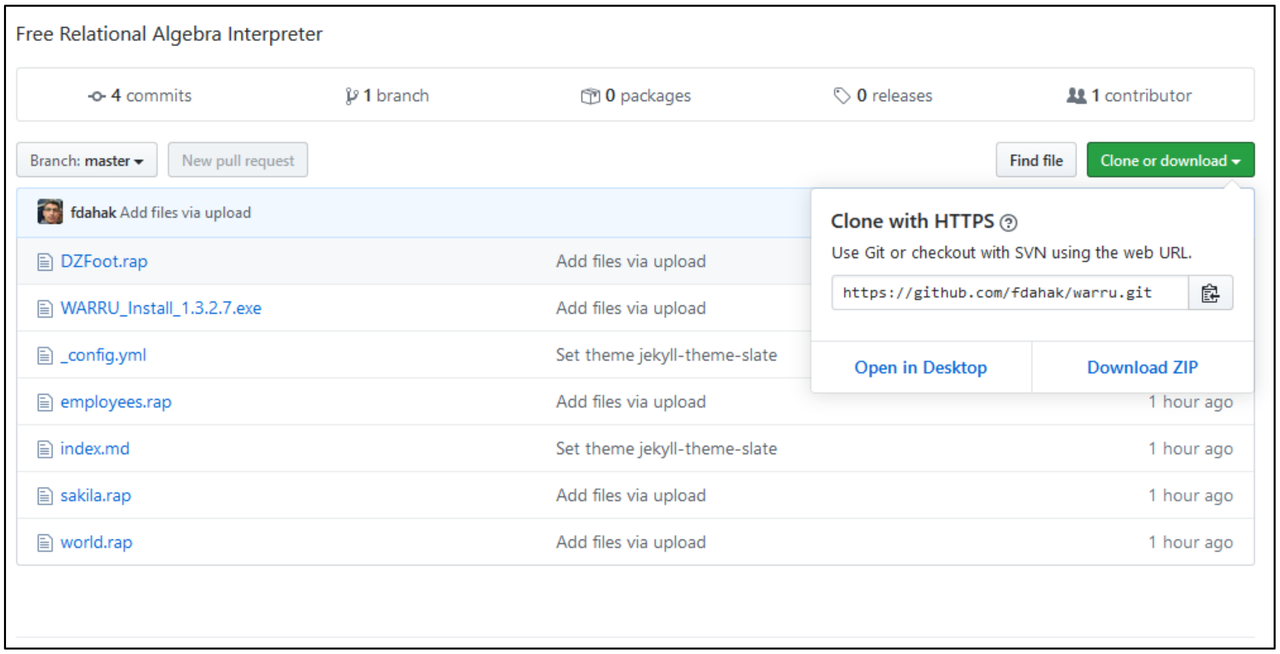
Warru is available on the following links:

http://dahak.esi.dz

https://github.com/fdahak/warru

Download the installation files and follow these instructions to install Warru on your computer.

Go to the following site https://github.com/fdahak/warru, click on the Clone Or Download link then click on the Download ZIP link.



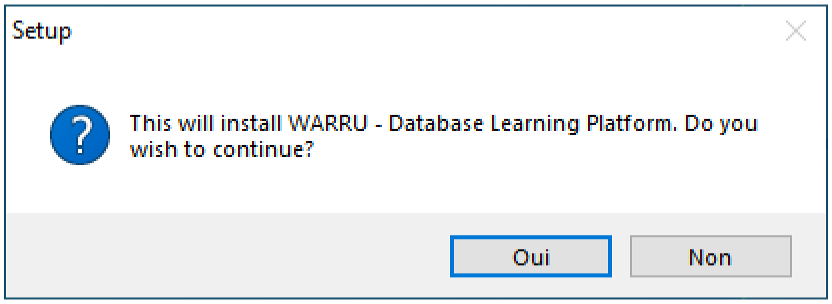
DZFoot.rap	Add files via upload	
WARRU_Install_1.3.2.7.exe	Add files via upload	
_config.yml	Set theme jekyll-theme-slate	
employees.rap	Add files via upload	1 hour ago
index.md	Set theme jekyll-theme-slate	1 hour ago
sakila.rap	Add files via upload	1 hour ago
world.rap	Add files via upload	1 hour ago

After downloading, unzip everything to a folder on your computer. Open the folder and you will find the following files:

File	Description
Warru_Install_1.3.2.7.exe	Warru setup file
User_Manual.pdf	This current user manual.
DZFoot.rap	Example of a Warru database containing the data of the Algerian football championship.
Sakila.rap	Example of a Warru database containing data from the Sakila database which comes with MySQL
World.rap	Example of a Warru database containing data on the countries of the world

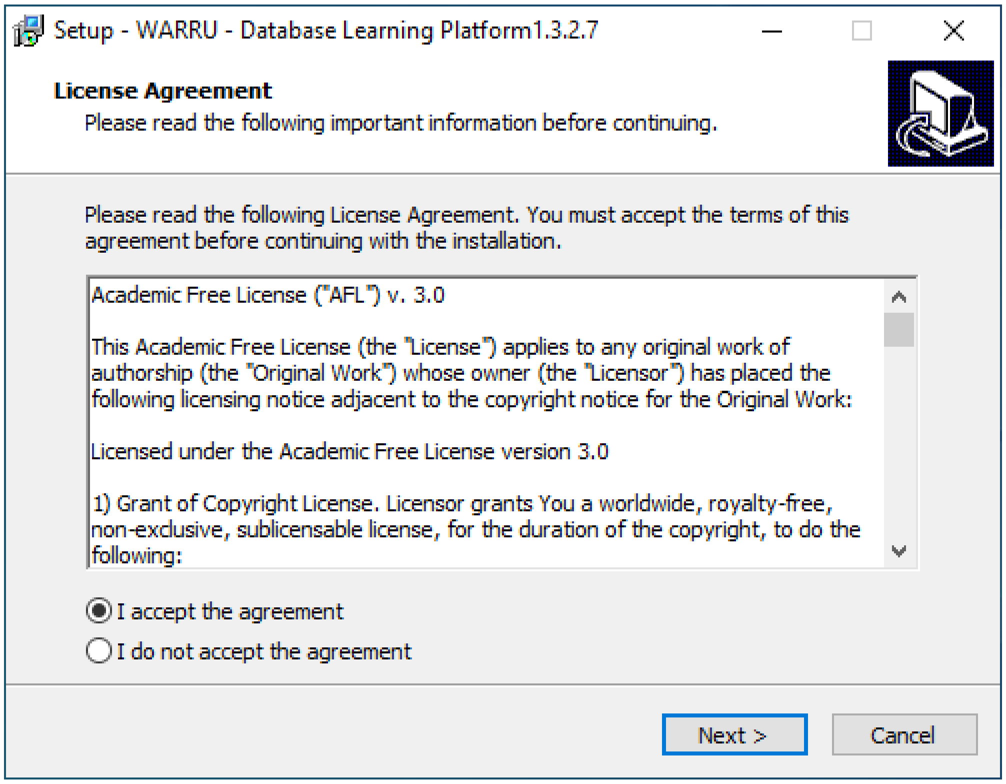
To install Warru in your computer, double click on the warru installation file then follow the steps below:

1. Confirm installation.


Setup

This will install WARRU - Database Learning Platform. Do you wish to continue?

2. Accept agreement declaration.



Setup - WARRU - Database Learning Platform1.3.2.7

License Agreement

Please read the following important information before continuing.

Please read the following License Agreement. You must accept the terms of this agreement before continuing with the installation.

Academic Free License ("AFL") v. 3.0

This Academic Free License (the "License") applies to any original work of authorship (the "Original Work") whose owner (the "Licensor") has placed the following licensing notice adjacent to the copyright notice for the Original Work:

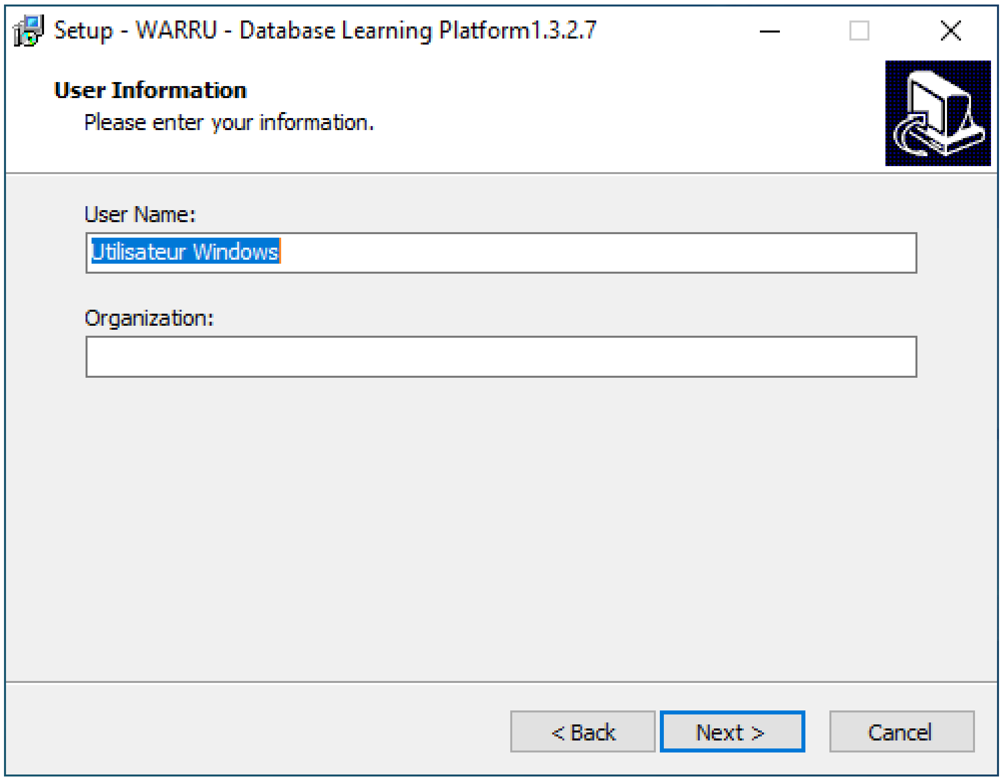
Licensed under the Academic Free License version 3.0

1) Grant of Copyright License. Licensor grants You a worldwide, royalty-free, non-exclusive, sublicensable license, for the duration of the copyright, to do the following:

● I accept the agreement

○ I do not accept the agreement

3. Insert username and organization.



Setup - WARRU - Database Learning Platform1.3.2.7

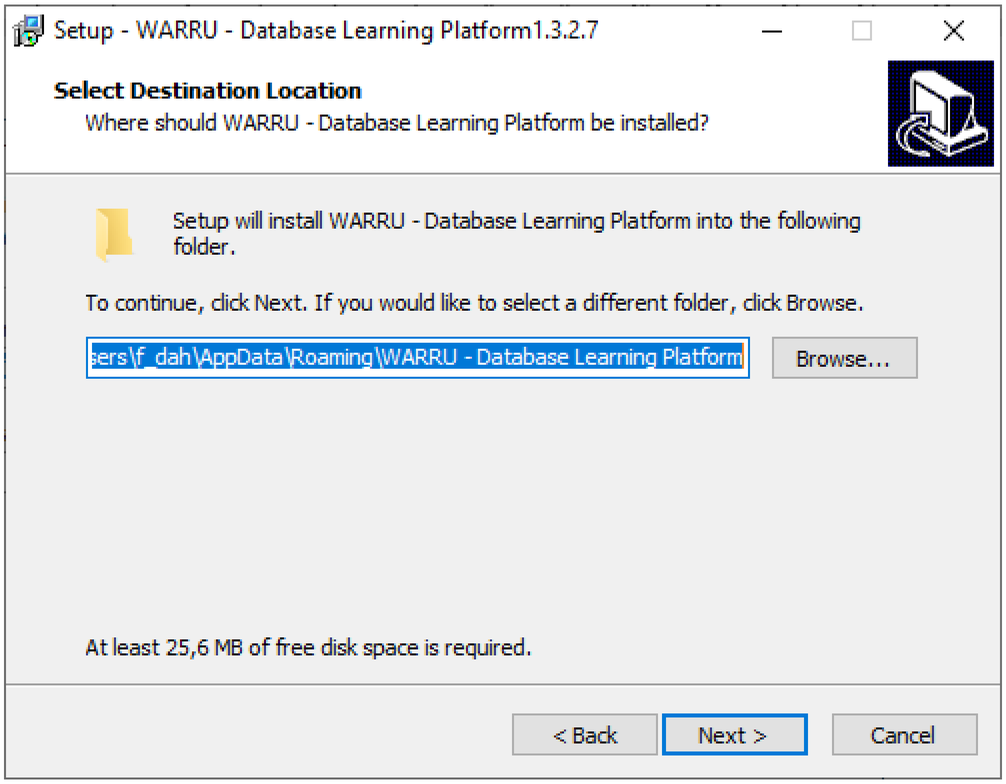
User Information

Please enter your information.

User Name: Utilisateur Windows

Organization:

4. Choose destination folder for your application.


Setup - WARRU - Database Learning Platform1.3.2.7

Select Destination Location

Where should WARRU - Database Learning Platform be installed?

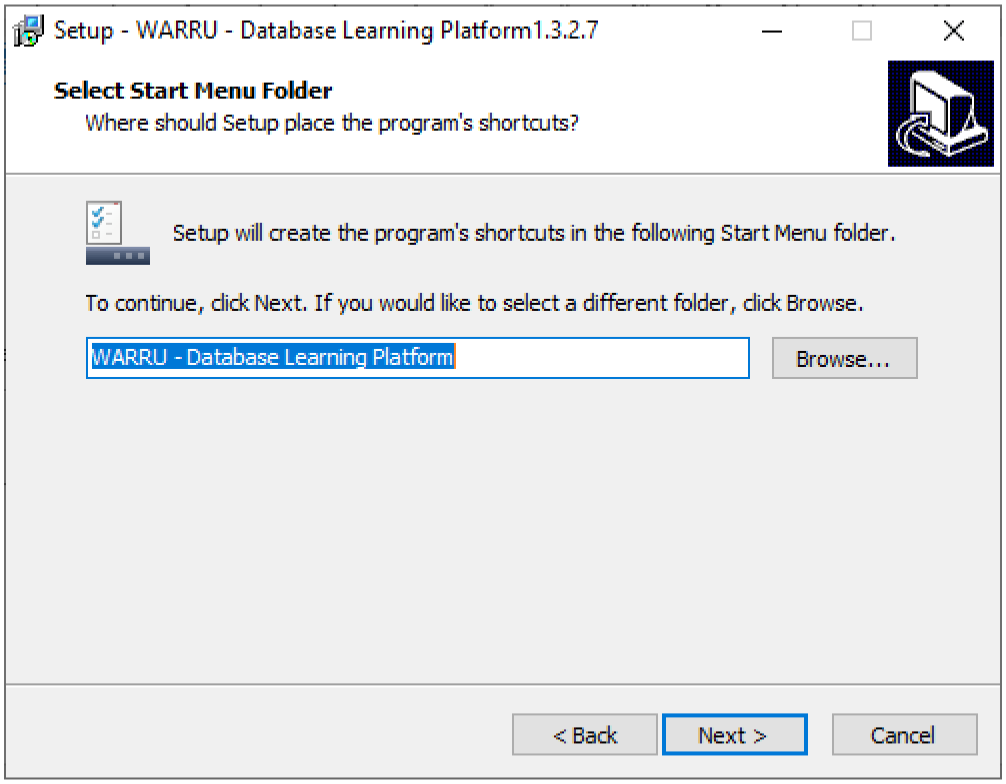
Setup will install WARRU - Database Learning Platform into the following folder.

To continue, click Next. If you would like to select a different folder, click Browse.

sers\f_dah\AppData\Roaming\WARRU - Database Learning Platform

At least 25,6 MB of free disk space is required.

5. Select a start menu folder.


Setup - WARRU - Database Learning Platform1.3.2.7

Select Start Menu Folder

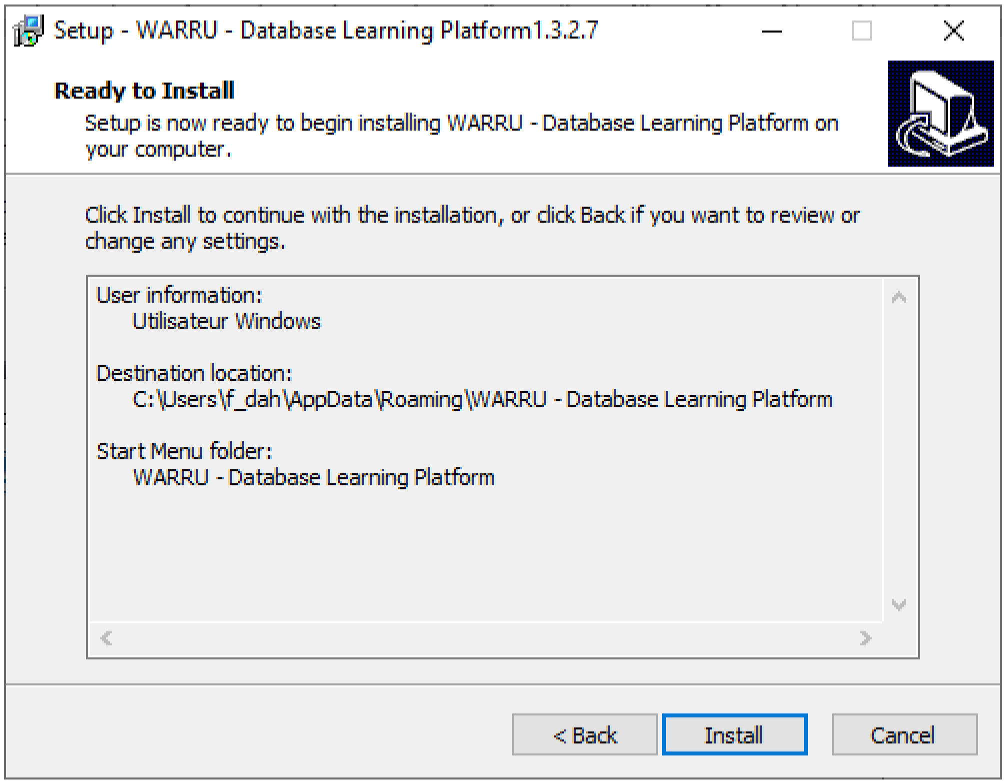
Where should Setup place the program's shortcuts?

Setup will create the program's shortcuts in the following Start Menu folder.

To continue, click Next. If you would like to select a different folder, click Browse.

WARRU - Database Learning Platform

6. Confirm by clicking Install button.



Setup - WARRU - Database Learning Platform1.3.2.7

Ready to Install

Setup is now ready to begin installing WARRU - Database Learning Platform on your computer.

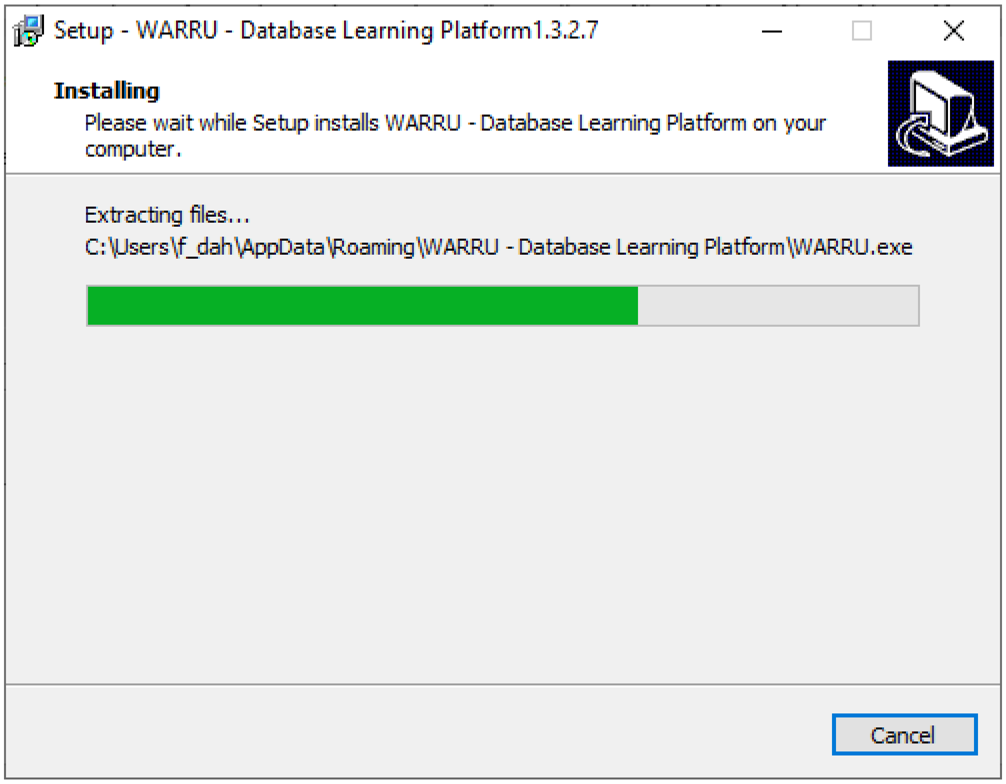
Click Install to continue with the installation, or click Back if you want to review or change any settings.

User information:
Utilisateur Windows

Destination location:
C:\Users\f_dah\AppData\Roaming\WARRU - Database Learning Platform

Start Menu folder:
WARRU - Database Learning Platform

7. Install in progress.



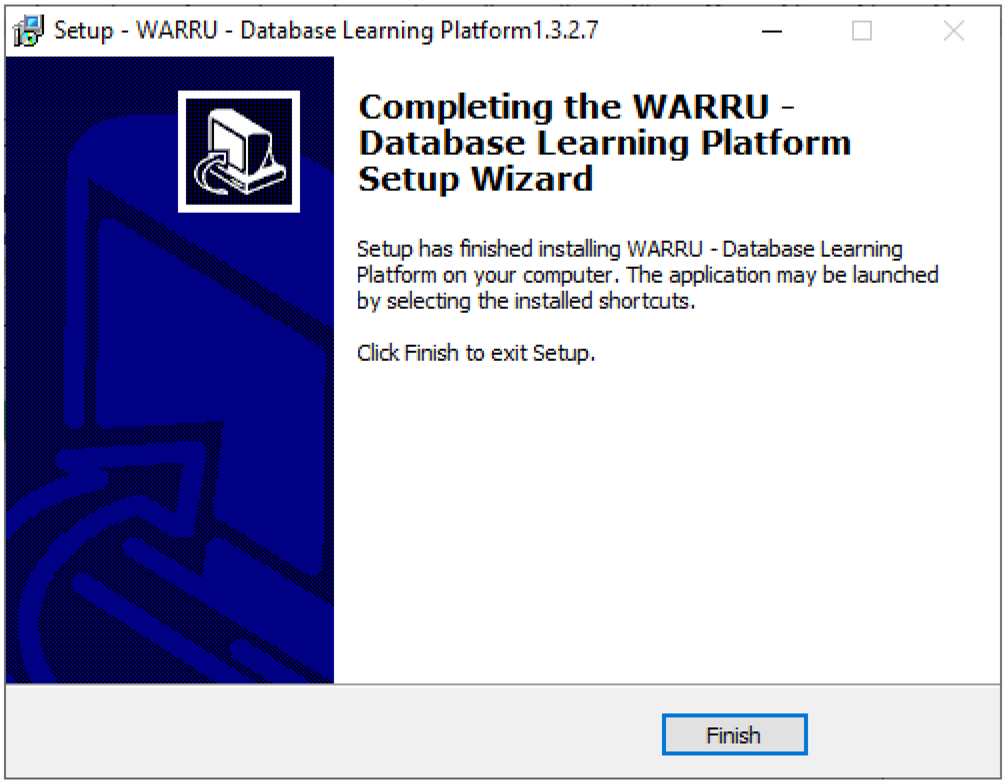
Setup - WARRU - Database Learning Platform1.3.2.7

Installing

Please wait while Setup installs WARRU - Database Learning Platform on your computer.

Extracting files...
C:\Users\f_dah\AppData\Roaming\WARRU - Database Learning Platform\WARRU.exe

8. Warru is now installed on your computer.



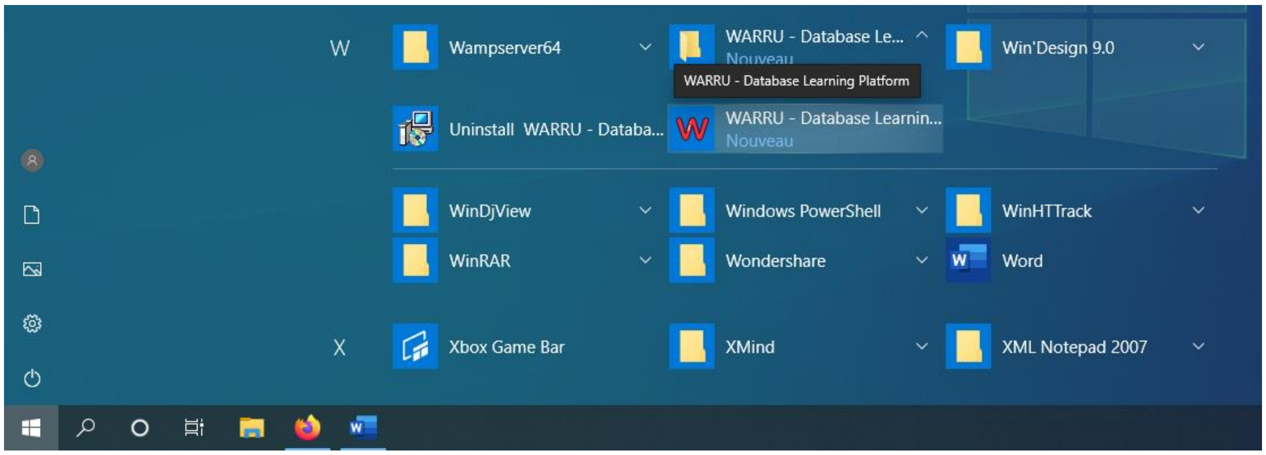
Setup - WARRU - Database Learning Platform1.3.2.7

Completing the WARRU - Database Learning Platform Setup Wizard

Setup has finished installing WARRU - Database Learning Platform on your computer. The application may be launched by selecting the installed shortcuts.

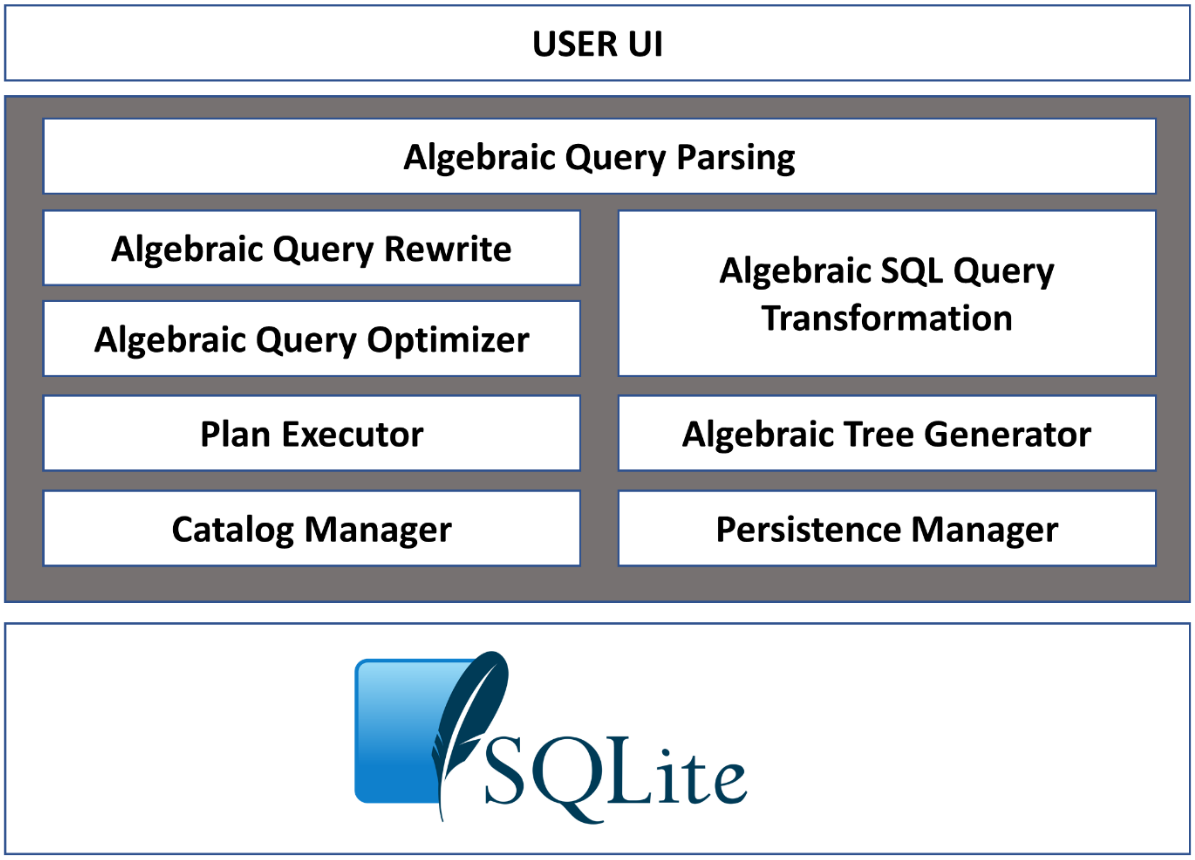
Click Finish to exit Setup.

To launch Warru just go to the Windows menu and click on the link "WARRU: Database Learning System"



III. Architecture

Warru is built on SQLite. The project is persisted in a SQLite database. It has an algebraic parser for compiling algebraic queries. The queries are then rewritten to be optimized by the query rewriter and optimizer modules. An algebraic query is then translated into SQL and then executed. Once the result has been produced, the algebraic tree is constructed.

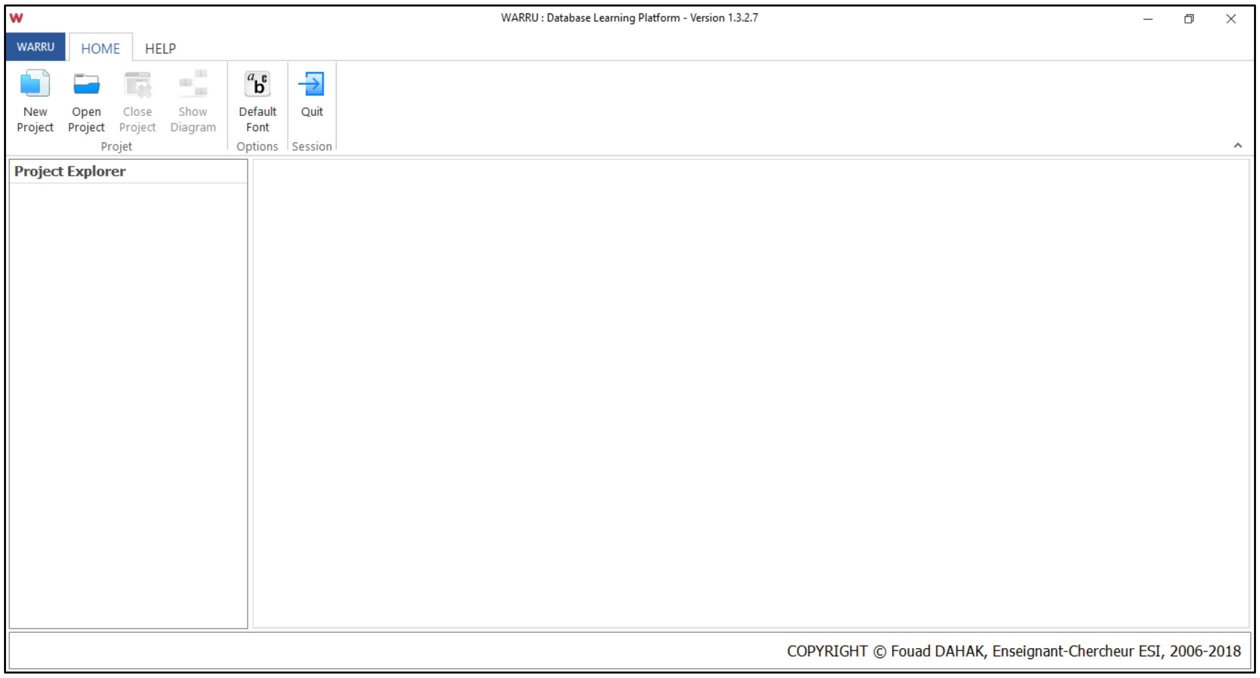


Each algebraic project is saved in a single .rap file which contains the user database and the system catalog containing configuration data, algebraic queries and SQL queries as well as SQL views.
The project is compressed, which makes it almost impossible to read with other tools.

IV. Getting started

1. User Interface

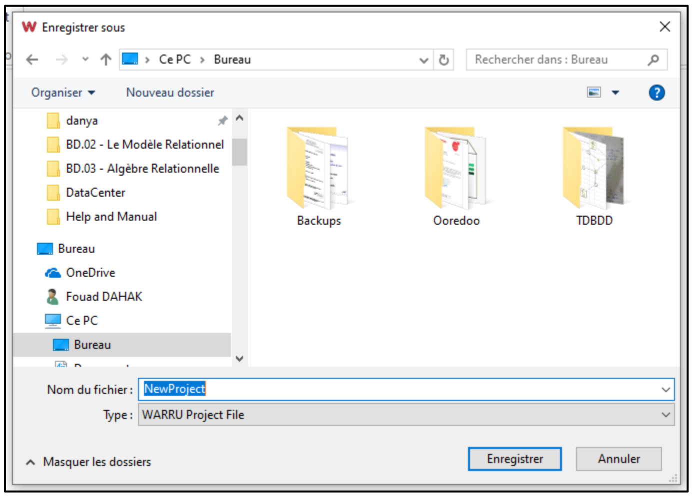
The main Warru window looks like this, a main toolbar at the top, a project inspector on the left and a work area in the center.



COPYRIGHT © Fouad DAHAK, Enseignant-Chercheur ESI, 2006-2018

2. New Project

To create a new project, click on the New Project button in the toolbar then insert the project name and the location of its source file.



Nom du fichier : NewProject
Type : WARRU Project File

Once a new project has been created or an existing project has been opened, the query manipulation toolbar is displayed. This toolbar allows you to perform all the necessary actions on queries and projects.

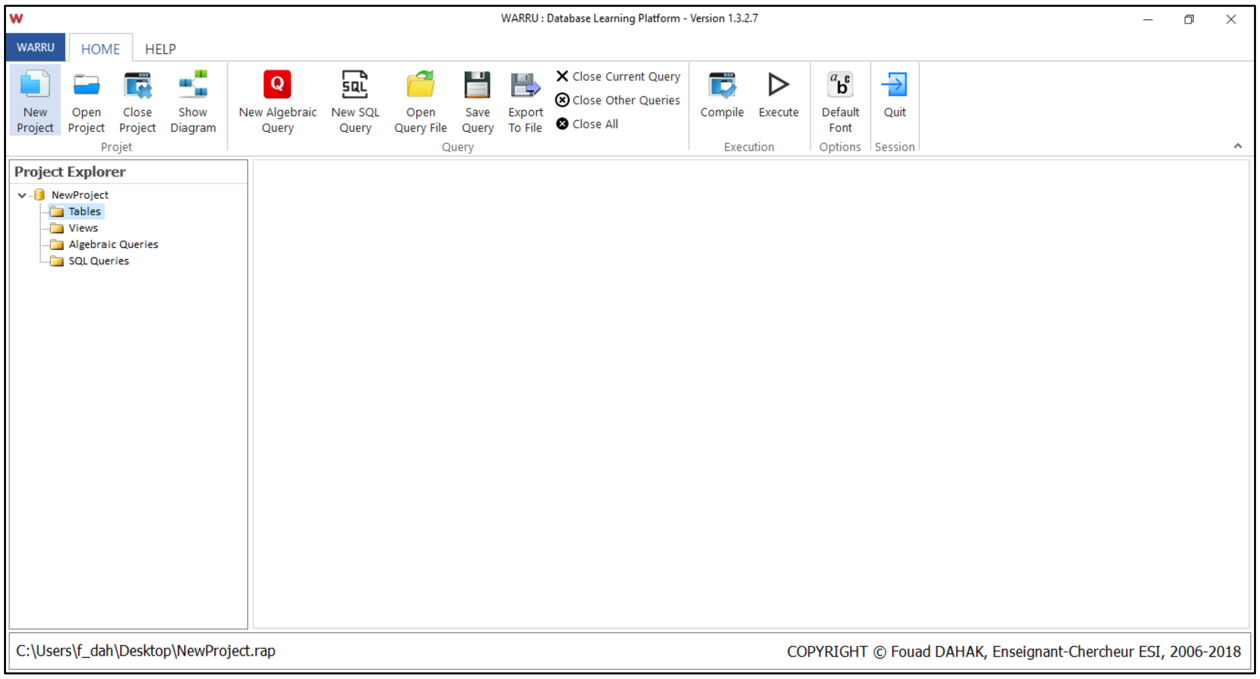
You will also be able to see the project structure displayed in the project explorer. Each Warru project contains the following elements:

Tables: All the tables created in the project.

Views: Set of SQL views created in the project.

Algebra Queries: All the algebraic queries saved in the project.

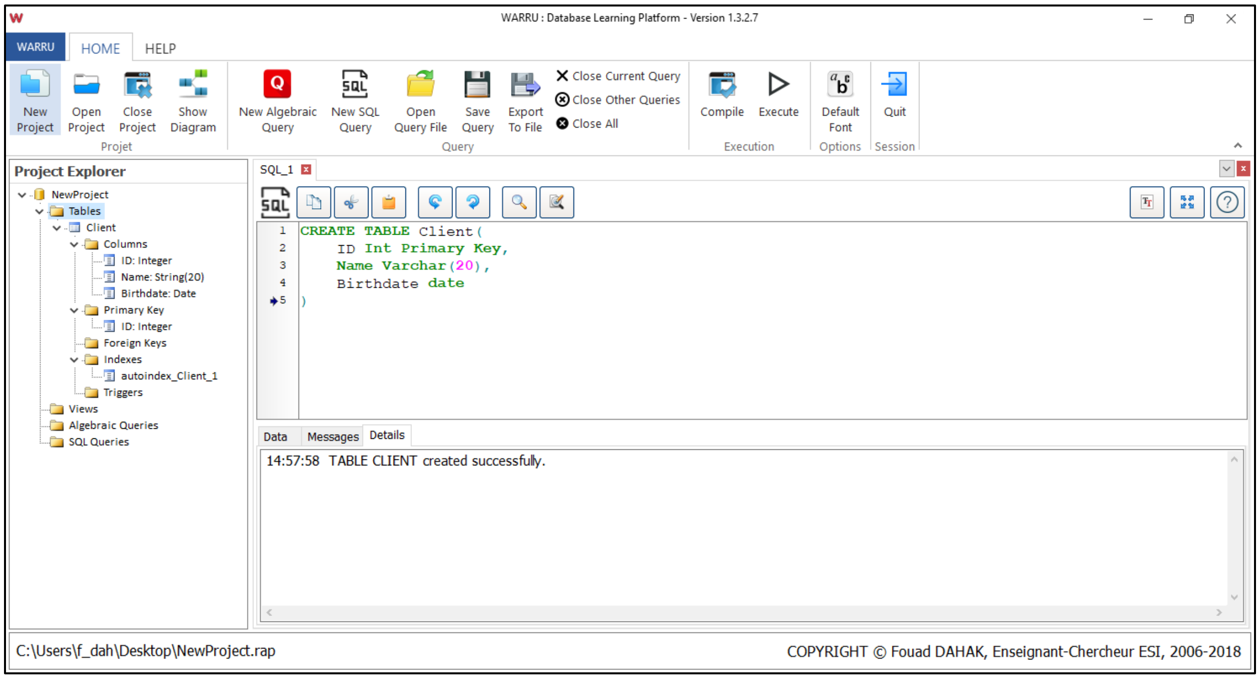
SQL Queries: Set of SQL queries saved in the project.



To add tables to the project you must use the SQL syntax by creating a new SQL query from the New SQL Query button.

Once the query has been executed with the Execute button, the result is immediately displayed in the project explorer.

Use the same procedure to create views and indexes on to alter tables.



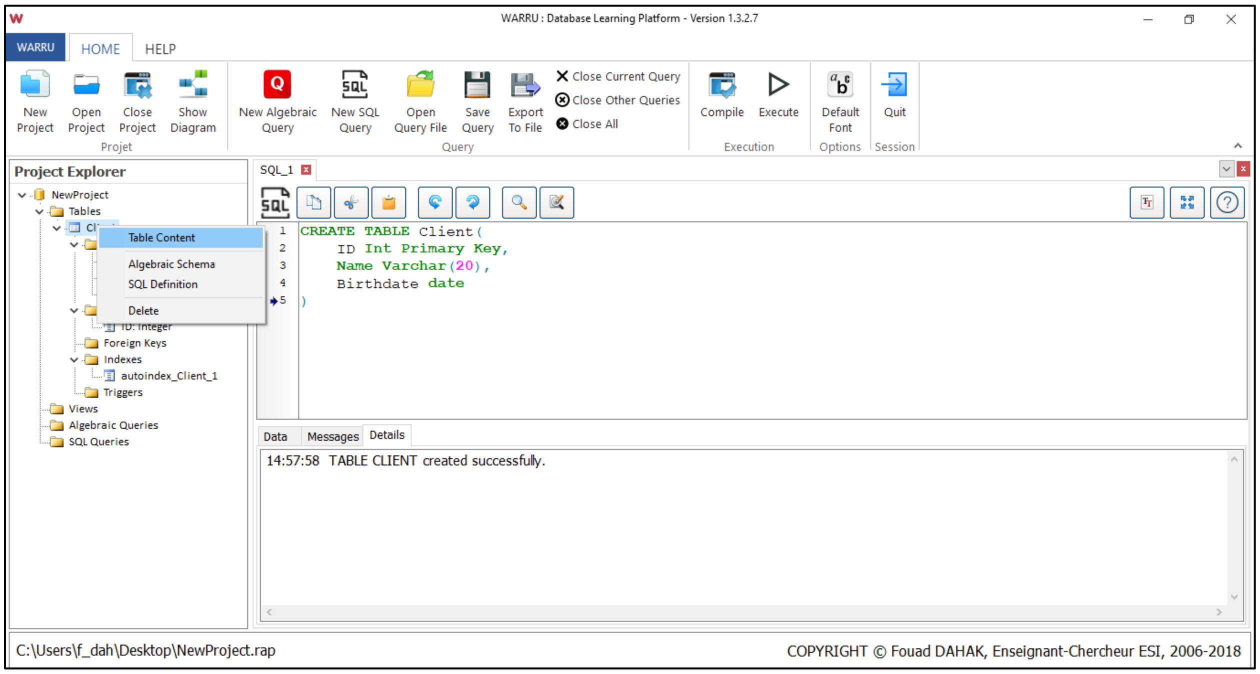
```
CREATE TABLE Client(
    ID Int Primary Key,
    Name Varchar(20),
    Birthdate date
)
```

```
14:57:58  TABLE CLIENT created successfully.
```

C:\Users\f_dah\Desktop\NewProject.rap

COPYRIGHT © Fouad DAHAK, Enseignant-Chercheur ESI, 2006-2018

By right clicking on a table, you can manipulate its data, display its SQL declaration as well as its algebraic declaration as you can delete it.



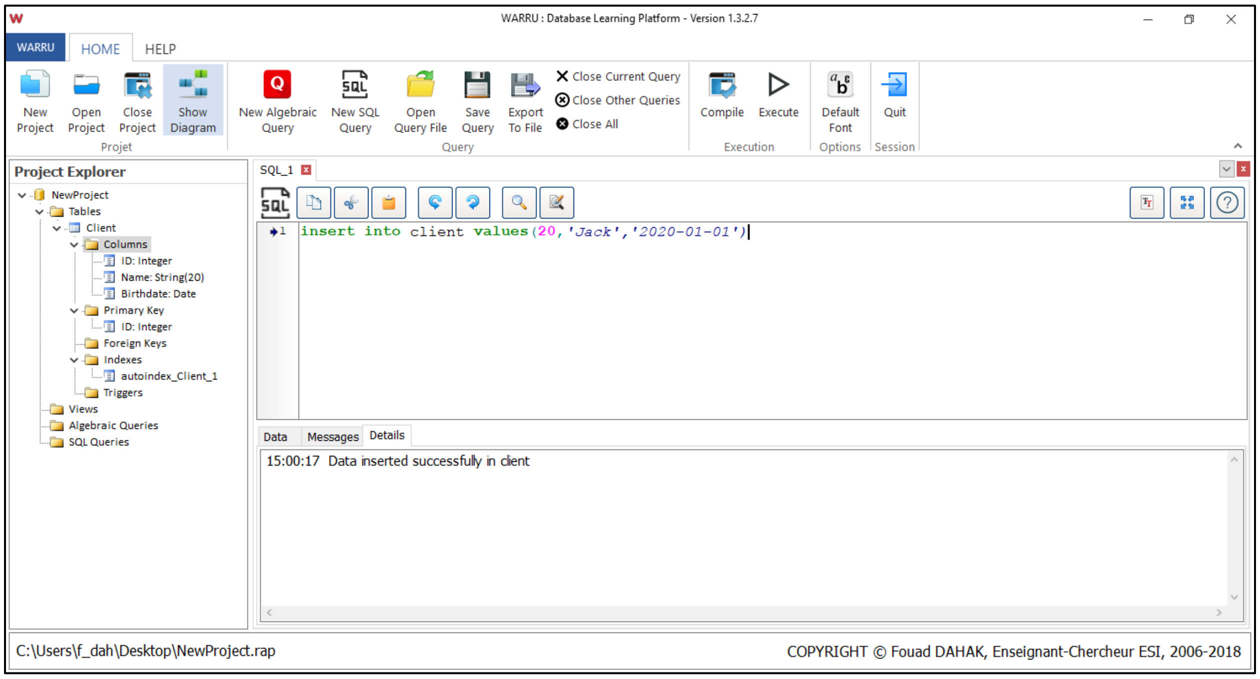
```
CREATE TABLE Client(
    ID Int Primary Key,
    Name Varchar(20),
    Birthdate date
)
```

```
14:57:58  TABLE CLIENT created successfully.
```

C:\Users\f_dah\Desktop\NewProject.rap

COPYRIGHT © Fouad DAHAK, Enseignant-Chercheur ESI, 2006-2018

Also use the SQL syntax to insert, modify or delete data from a table.

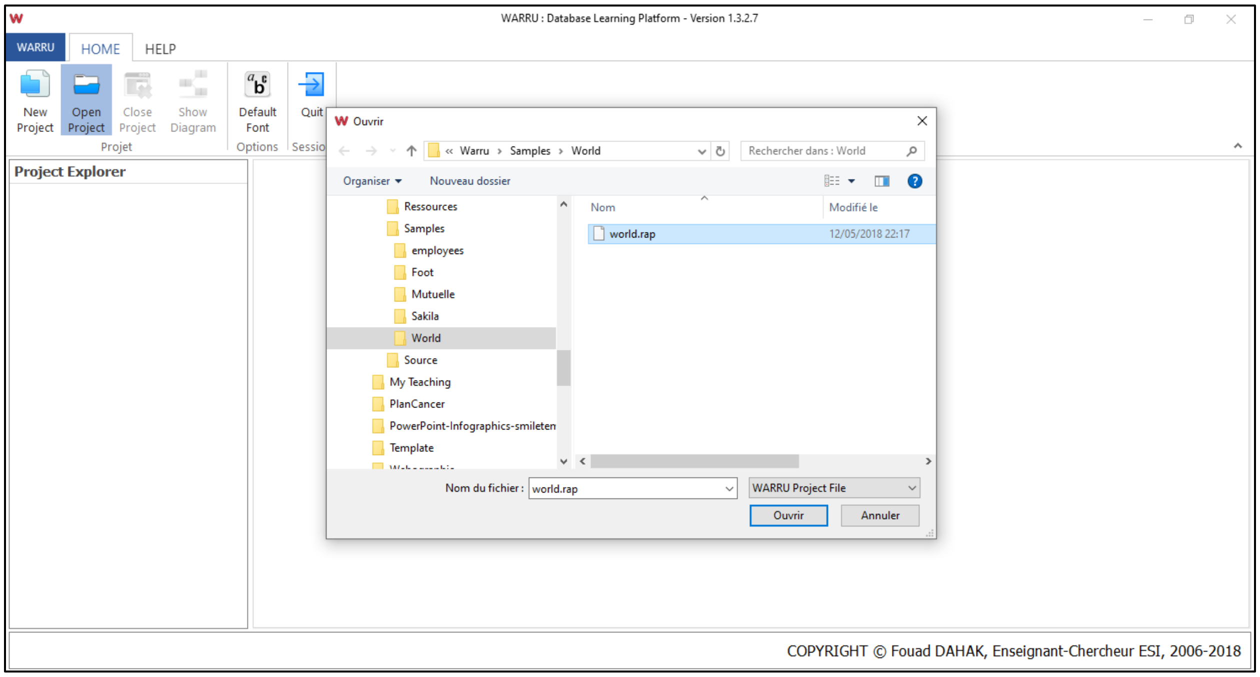


```
insert into client values(20,'Jack','2020-01-01')
```

```
15:00:17  Data inserted successfully in client
```

3. Open Existing Project

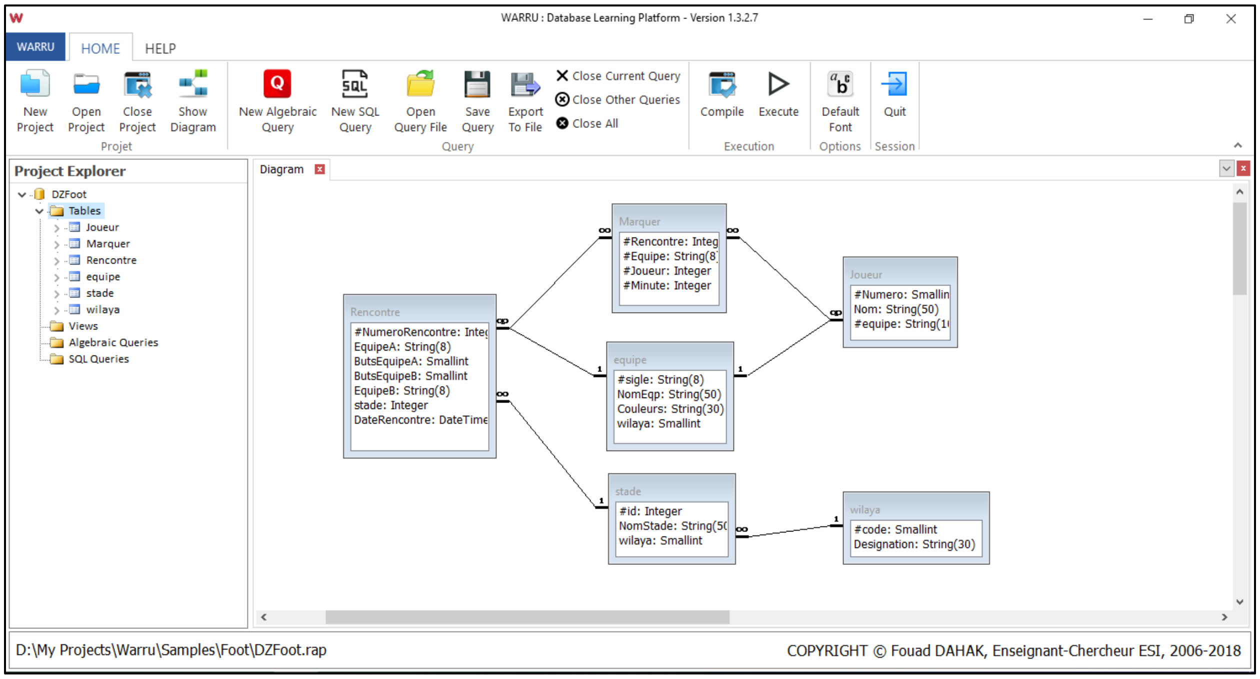
To open an existing project, go to the toolbar and click on the open project button.



Nom du fichier : world.rap

Once the project is open you can see its structure in the project explorer.

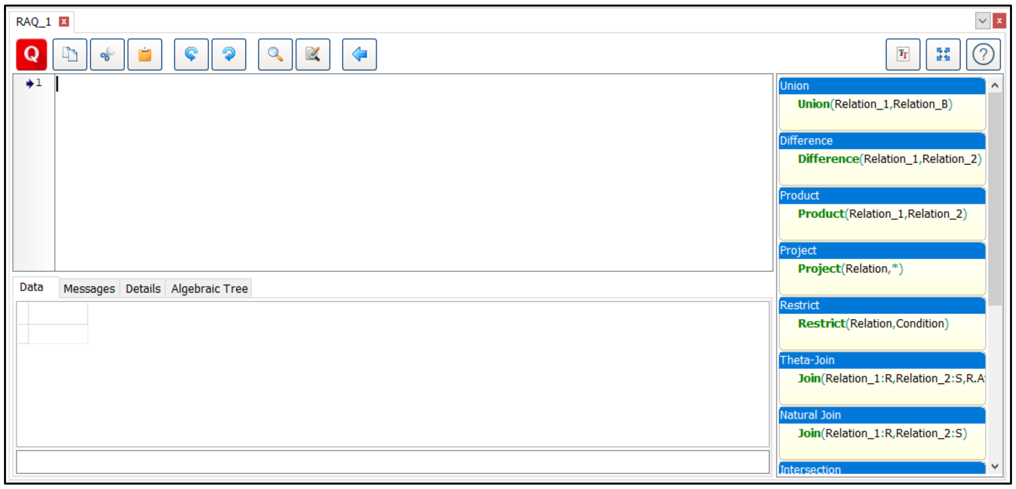
To display the database diagram, click on the Show Diagram button. Once displayed, you can rearrange the tables positions, save the diagram and export it as an image.



4. Query Editor

4.1. Query Editor

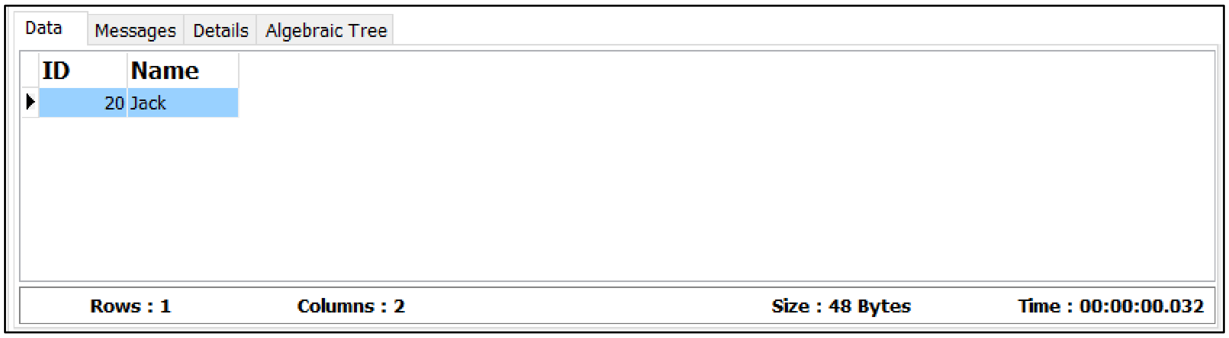
Warru has a powerful query editor allowing you to build both algebraic and SQL queries. The editor also has a query templates panel allowing you to quickly build your queries. To do this, simply drag the template into the query editing area to have the corresponding algebraic or SQL code.


RAQ_1
Data
Messages
Details
Algebraic Tree
Union
Union(Relation_1,Relation_B)
Difference
Difference(Relation_1,Relation_2)
Product
Product(Relation_1,Relation_2)
Project
Project(Relation,*)
Restrict
Restrict(Relation,Condition)
Theta-Join
Join(Relation_1:R,Relation_2:S,R.A
Natural Join
Join(Relation_1:R,Relation_2:S)
Intersection

4.2. Query Parsing

Once the query has been written, you have the option of compiling or executing it directly. In the event of errors, messages are displayed in the Messages area at the bottom of the editor.

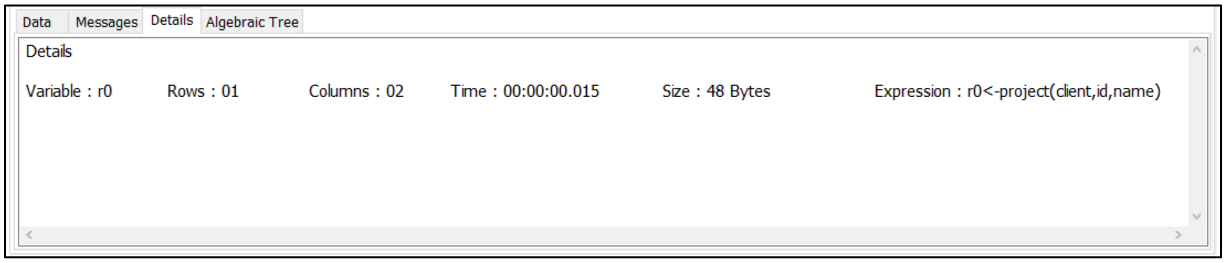
When executed correctly, the query result is displayed in the results grid.



ID	Name
20	Jack

Rows : 1 Columns : 2 Size : 48 Bytes Time : 00:00:00.032

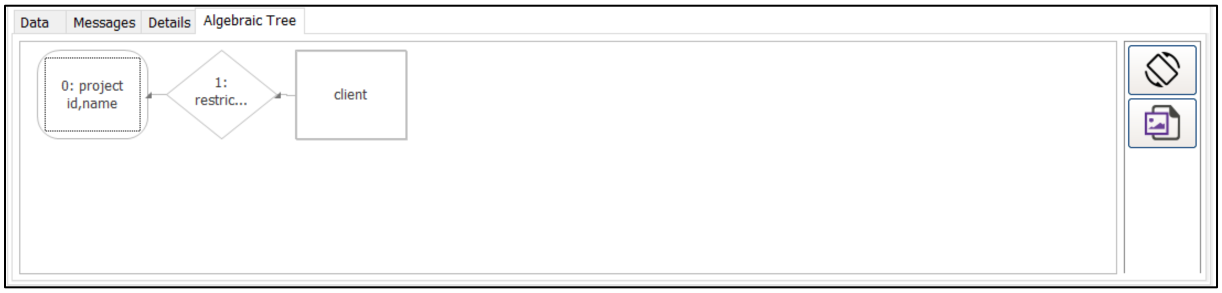
A detailed execution plan is displayed in the execution plan tab containing for each sub-query its syntax, its execution time and the consumed memory.



Details

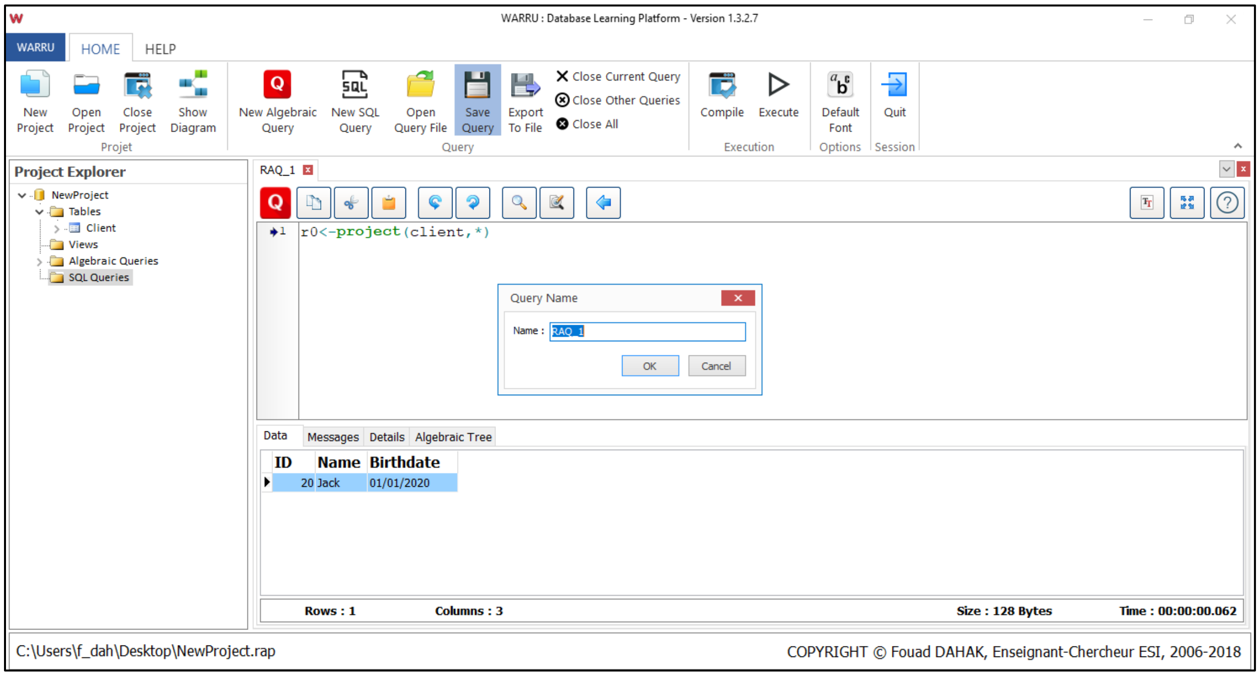
Variable : r0 Rows : 01 Columns : 02 Time : 00:00:00.015 Size : 48 Bytes Expression : r0<-project(client,id,name)

A graphical algebraic tree is also constructed for each query executed. The tree can be exported as a PNG image.



4.3. Saving and exporting queries

To save a query in the project, click on the Save Query button then give a name for the query and click on save. Once saved, the query appears in the project explorer. To open it just double click on it.



```
r0<-project(client,*)
```

ID	Name	Birthdate
20	Jack	01/01/2020

Rows : 1 Columns : 3 Size : 128 Bytes Time : 00:00:00.062

C:\Users\f_dah\Desktop\NewProject.rap

COPYRIGHT © Fouad DAHAK, Enseignant-Chercheur ESI, 2006-2018

To delete a query, right click on it in the project explorer then click on the delete link.

V. Algebraic Language

Warru is a powerful algebraic editor allowing you to perform any type of algebraic query. All algebraic operations are implemented and supported.

1. Algebraic Operation Template

The standard form of an algebraic operation is as follows:

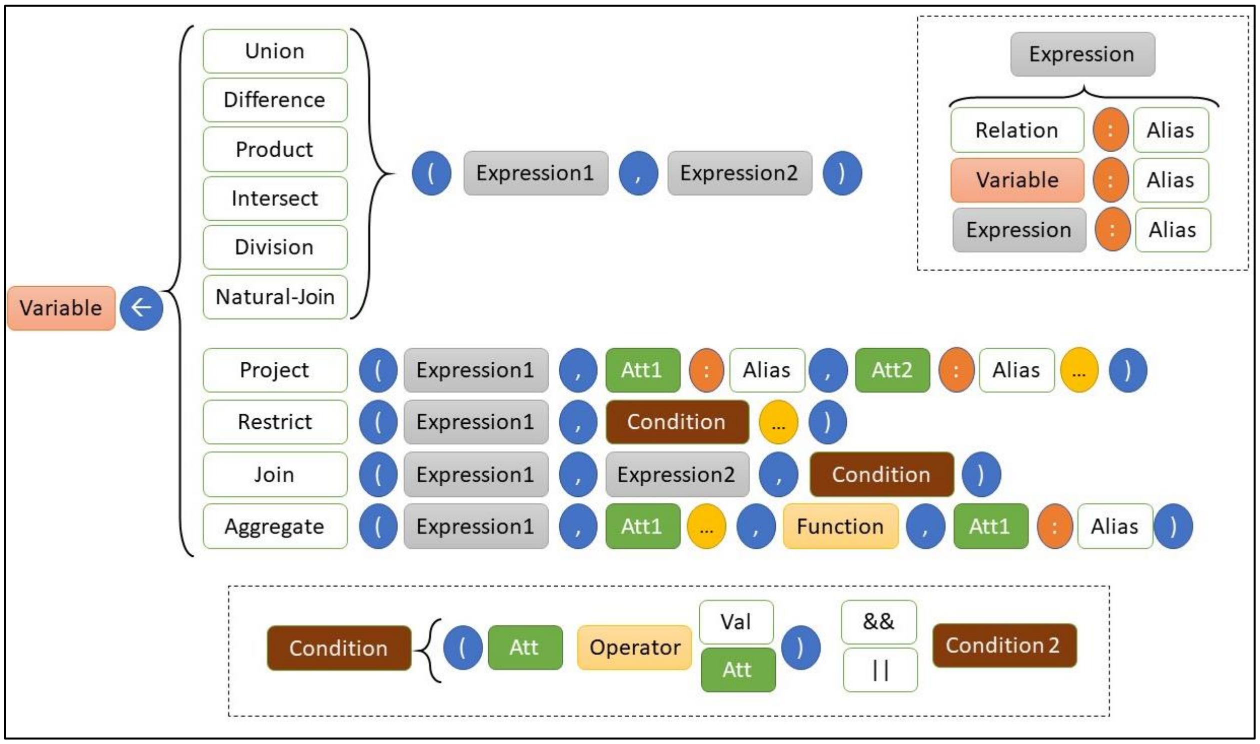
$$\textbf{Variable} \leftarrow \textbf{Operation\_Name}(\text{Expression}, \text{Param}_1, \ldots, \text{Param}_n)$$

Such as ***Variable*** is the variable name (Do not use special chars when naming variables). ***Expression*** can be a relation, a variable or an algebraic expression. ***Param$_i$*** is a list of the operation parameters.

2. Algebraic Operation Syntax

2.1. Operations

The figure below summarizes all the algebraic operations supported in Warru.


Union
Difference
Product
Intersect
Division
Natural-Join
(Expression1 , Expression2)
Expression
Relation : Alias
Variable : Alias
Expression : Alias
Variable ←
Project (Expression1 , Att1 : Alias , Att2 : Alias ...)
Restrict (Expression1 , Condition ...)
Join (Expression1 , Expression2 , Condition)
Aggregate (Expression1 , Att1 ... , Function , Att1 : Alias)
Condition (Att Operator Val Att) && || Condition 2

The table below summarizes the algebraic operations supported in Warru. The syntax of each of them is given in addition to a brief description.

2.1.1. Union

Variable ←**Union**(Expression1,Expression2)

Union is used to integrate results from different sources but with the same structure. These sources can be relations or algebraic expressions.

2.1.2. Difference

Variable ←Difference(Expression1,Expression2)

Difference is used when looking for elements that have performed actions and have not performed other actions. It is often used when looking for elements that did not participate in facts because often it is the facts that are saved.

2.1.3. Product

```
Variable ←Difference(Expression1,Expression2)
```

Cartesian product is an operation which is not very useful on its own. It is often combined with other algebraic operations to give a significant result.

2.1.4. Project

```
Variable ←Project(Expression,Att1, Att2,…,Attn)
```

Projection is one of the important operations of relational algebra. It displays only the desired columns of a given table.

2.1.5. Restrict

```
Variable ←Restrict(Expression, Condition)
```

Restriction is an operation used to perform filters on the data. Like projection, it is very important for building algebraic queries.

2.1.6. Theta-Join

```
Variable ←JOIN(Expression1,Expression2, Condition)
```

Theta-join makes it possible to make the link between several tables with a condition between the attributes of these tables. The condition must be constructed correctly to guarantee a correct result.

2.1.7. Natural Join

```
Variable ←JOIN(Expression1,Expression2)
```

Natural join works in the same way as the theta-join, except that the condition is constructed automatically and is represented as having equality between all the attributes of the two tables having the same name. It also displays these attributes without duplication in the result.

2.1.8. Intersect

```
Variable ←INTERSECT(Expression1,Expression2)
```

Intersection makes it possible to have the elements having participated in two facts at the same time.

2.1.9. Division

```
Variable ←DIVISION(Expression1,Expression2)
```

When one does not know the value with which an attribute is compared in a restriction condition or when these values are not a scalar but a table. In this case, division is used instead of a cascade use of several restrictions. Division is often used in aggregate queries.

2.1.10. Outer Join

```
Variable ←XJOIN(Expression1,Expression2, Condition)
Variable ←XJOIN(Expression1,Expression2)
```

Outer join, whether theta or natural, allows a result composed of all the rows of a table but only the rows satisfying a given condition of the second table.

2.1.11. Semi-Join

Variable ←SJOIN(Expression1,Expression2, Condition)

Variable ←SJOIN(Expression1,Expression2)

Semi-join is a simple join showing only the columns of the specified table.

2.1.12. Aggregates

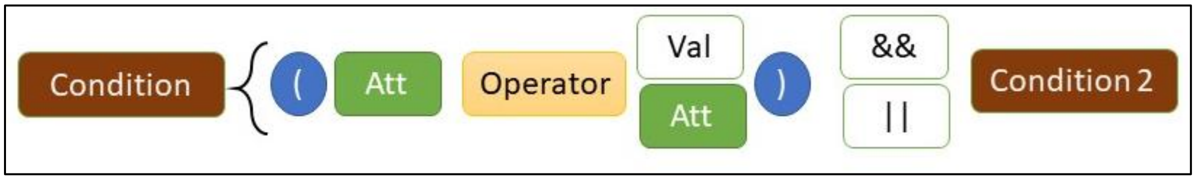
Variable ←AGREGAT(Expression, FONCTION, Attr)

Variable ←AGREGAT(Expression1,GroupByAttrinutes,Function,Att)

Aggregates are used to perform calculations on the entire columns of a table. They are used to calculate statistics such as the row number, maximum or minimum value, the average and the sum. These statistics can also be grouped by so-called grouping attributes.

2.2. Conditions

Conditions are used in restrictions or joins. they have the following forms:



<u>Attribute OPERATOR Value</u> Or Attribute *<u>OPERATOR Attribute</u>*

The allowed arithmetic operators in a condition are:

'=' equal to

'<' Less than

'<=' Less than or equal to

'>' Greater than

'> =' Greater than or equal to

'<>' Different from

The logical operators are:

'NOT' for negation

'&&' logical AND between two conditions

'||' logical OR between two conditions

2.3. Aliases

Use of aliases makes it possible to rename the columns in the query result or to simplify the algebraic expressions names in the query.

The symbol used for the alias declaration is '**:**'.

To rename the columns we use it in the following cases:

Project (Expression, AttributeName: Alias)

Agregat (Expression, Function, Attribute: Alias)

Agregat (Expression, GroupAttribute:Alias1,Function, Attribute: Alias2)

For the simplifications of the names of the expressions we use them in the following case:

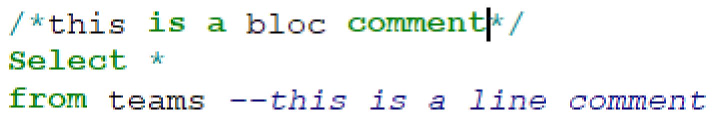
Join (Expression1 : Alias1, Expression2: Alias2)

2.4. Comments

You can add comments anywhere in an algebraic query by using the symbol '//' and '--' in a SQL query, anything after the comment symbol is ignored when the query is compiled.

```
// 09: What is the best defense?
r1<-Agregat(match,teama_id:team_id,sum,goals_teamb:goals)// sum of goals received by team A
r2<-Agregat(match,teamb_id:team_id,sum,goals_teama:goals)// sum of goals received by team B
r3<-Union(r1,r2)
```

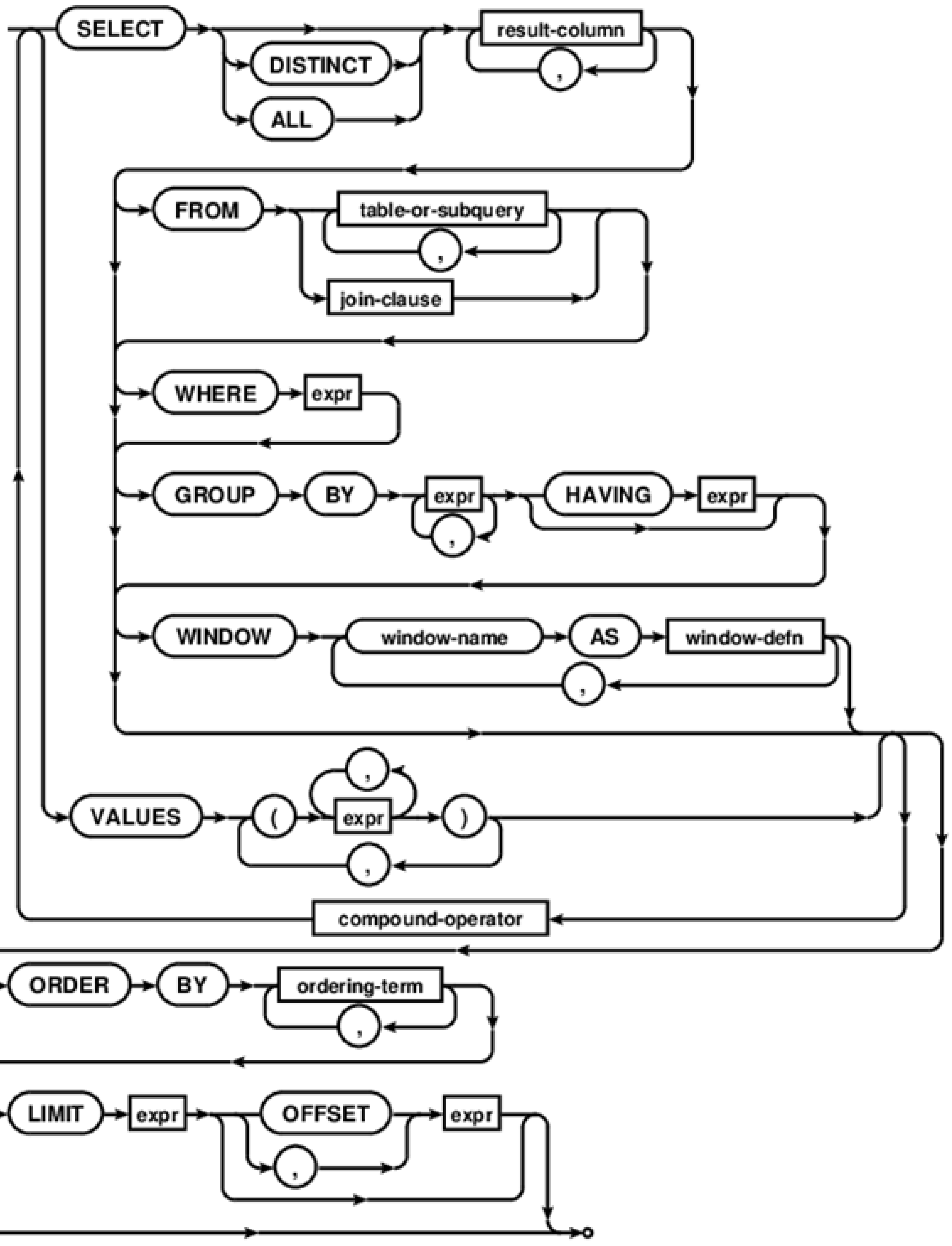
You can also add block comments in SQL queries by using the symbols /* and */



```
/*this is a bloc comment*/
Select *
from teams --this is a line comment
```

VI. SQL Queries

Warru fully supports SQL 92 as well as all the features implemented in SQLite. You can therefore formulate any queries in standard SQL.

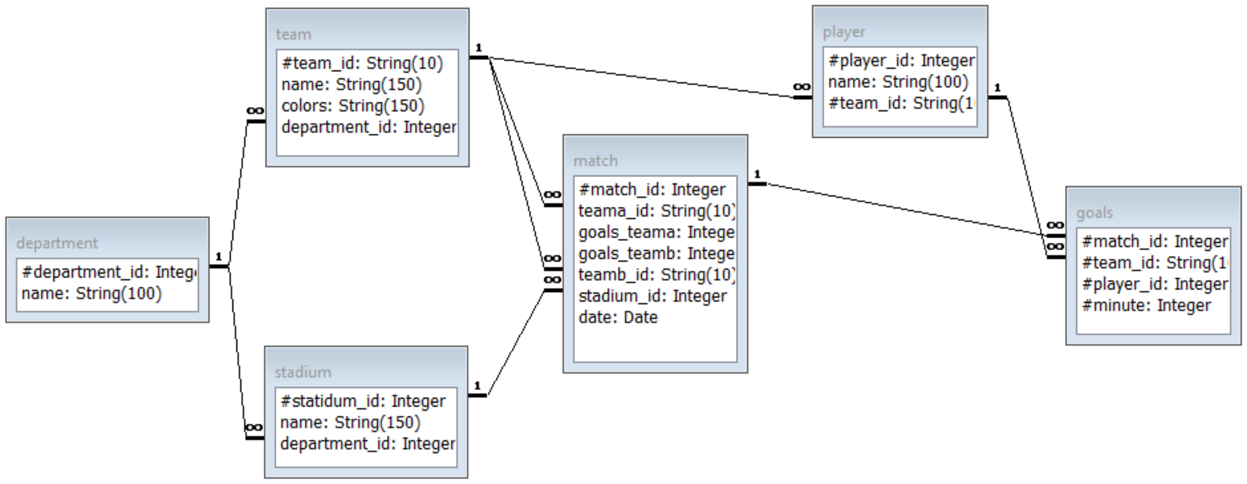


VII. Samples

Warru contains three sample projects, each containing real data with algebraic and SQL queries. These examples will allow students to learn algebraic language and SQL as long as each query is provided with its answer.

1. DZFoot

DZFoot is a database containing the data relating to the first league of the Algerian football championship. It contains all the teams, players, matches, stadiums and even the goals scored in the different matches. A set of queries with their solutions is also provided.

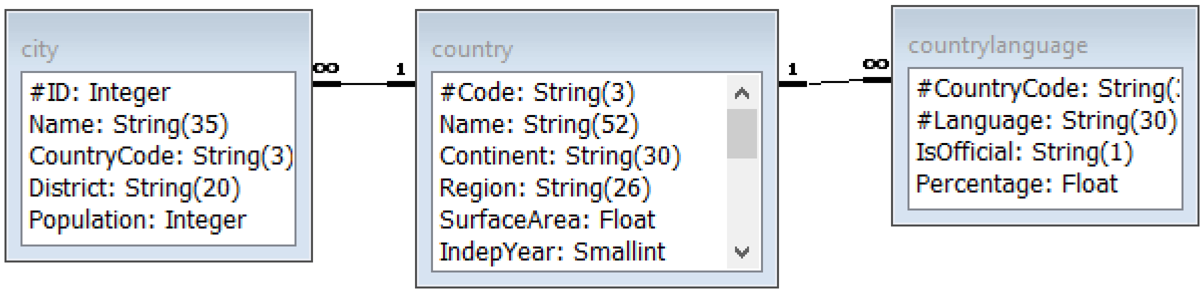


2. Sakila

Sakila is the famous movie rental database integrated as a sample database in MySQL. A set of queries has also been integrated into this database.

3. World

World is a database containing information on countries around the world, their cities and the languages spoken in these countries. Very interesting queries have also been integrated into this database.


city
#ID: Integer
Name: String(35)
CountryCode: String(3)
District: String(20)
Population: Integer
country
#Code: String(3)
Name: String(52)
Continent: String(30)
Region: String(26)
SurfaceArea: Float
IndepYear: Smallint
countrylanguage
#CountryCode: String(
#Language: String(30)
IsOfficial: String(1)
Percentage: Float
∞
1