

Interpréteur d'algèbre relationnelle

Université Mons Hainaut
Première licence en informatique de gestion
Directeur du projet : M. Jef Wijzen

Olivier Christiaen

Septembre 2004

Plan de la présentation

- ◆ **Objectif du projet**
- ◆ **L'algèbre relationnelle**
- ◆ **Critères de fonctionnement**
- ◆ **Développement**
- ◆ **Conclusions**

L'objectif du projet est de réaliser un **outil didactique** qui permettra aux étudiants en bases de données de **saisir une requête en algèbre relationnelle** et d'en voir le résultat.



- ◆ L'algèbre relationnelle est une méthode d'extraction permettant la manipulation des tables (ou relations).
- ◆ Son principe repose sur la création de nouvelles tables (tables résultantes).
- ◆ 6 opérations de base : sélection, projection, jointure, renommer, union, différence.

Algèbre relationnelle > Sélection

La sélection produit, à partir d'une relation, une relation résultante de même schéma mais ne comportant que les tuples qui répondent à la condition précisée en argument.

En sql : `@let CCOURS:COURS WHERE Prof=Mens`

<i>COURS</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4
	Bases de données	Wijsen	10
	Analyse	Troestler	10
	Data Mining	Wijsen	10

<i>CCOURS</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4

Algèbre relationnelle > Projection

La projection produit, à partir d'une relation, une relation résultante de schéma différent en ne conservant de la relation initiale que les attributs mentionnés en opérandes et les tuples correspondants en éliminant les doublons éventuels.

En sql : `@let PROF:COURS PROJECT Prof`

<i>COURS</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4
	Bases de données	Wijsen	10
	Analyse	Troestler	10
	Data Mining	Wijsen	10

<i>PROF</i>	<i>Prof</i>
	Mens
	Wijsen
	Troestler

Algèbre relationnelle > Jointure

La jointure produit, à partir de deux relations R1 et R2, une relation résultante de schéma différent dont les tuples sont obtenus en composant un tuple de R1 et de R2 lorsque ceux-ci ont la même valeur d'attribut pour des attributs de même nom. S'il n'y a pas d'attribut en commun, l'effet de la jointure est un produit cartésien.

En sqf : `@let DETAILS:COURS JOIN NOTES`

<i>COURS</i>	<i>Nom cours</i>	<i>Prof</i>	<i>Ects</i>	<i>NOTES</i>	<i>Etud</i>	<i>Nom cours</i>	<i>Note</i>
	Gestion de projets	Mens	4		Ed	Gestion de projets	13
	Bases de données	Wijsen	10		Ed	Bases de données	15
	Analyse	Troestler	10		Tim	Bases de données	13
	Data Mining	Wijsen	6		Eric	Gestion de projets	19

<i>DETAILS</i>	<i>Nom cours</i>	<i>Prof</i>	<i>Ects</i>	<i>Etud</i>	<i>Note</i>
	Gestion de projets	Mens	4	Ed	13
	Gestion de projets	Mens	4	Eric	19
	Bases de données	Wijsen	10	Ed	15
	Bases de données	Wijsen	10	Tim	13

Algèbre relationnelle > Renommer

Renommer produit, à partir d'une relation, une relation résultante dans laquelle l'attribut désigné a été renommé par la nouvelle valeur transmise.

En sqf :

```
@let CCOURS:COURS RENAME Nom AS Nom_cours
```

<i>COURS</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4
	Bases de données	Wijsen	10
	Analyse	Troestler	10
	Data Mining	Wijsen	10

<i>CCOURS</i>	<i>Nom_cours</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4
	Bases de données	Wijsen	10
	Analyse	Troestler	10
	Data Mining	Wijsen	10

Algèbre relationnelle > Union

L'union produit, à partir de deux relations de même schéma, une relation résultante de même schéma ayant pour tuples ceux appartenant au deux ou à une des relations en éliminant les doublons éventuels.

En sqf :

```
@let C2000_2001:COURS_2000 UNION COURS_2001
```

<i>COURS_2000</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>	<i>COURS_2001</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4		Gestion de projets	Bruyère	4
	Bases de données	Wijsen	10		Bases de données	Wijsen	10
	Analyse	Troestler	10		Analyse	Troestler	10
	Data Mining	Wijsen	6		Data Mining	Wijsen	6
					OMT	Goossens	10

<i>C2000_2001</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4
	Bases de données	Wijsen	10
	Analyse	Troestler	10
	Data Mining	Wijsen	6
	Gestion de projets	Bruyère	4
	OMT	Goossens	10

Algèbre relationnelle > Différence

La différence produit, à partir de deux relations de même schéma, une relation résultante de même schéma ayant pour tuples ceux appartenant à la première relation mais pas à la seconde.

En sqf : `@let DIFF_C :Cours_2001 MINUS Cours_2000`

<i>COURS_2000</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4
	Bases de données	Wijsen	10
	Analyse	Troestler	10
	Data Mining	Wijsen	6

<i>COURS_2001</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Bruyère	4
	Bases de données	Wijsen	10
	Analyse	Troestler	10
	Data Mining	Wijsen	6
	OMT	Goossens	10

<i>DIFF_C</i>	<i>Nom</i>	<i>Prof</i>	<i>Ects</i>
	Gestion de projets	Mens	4

Algèbre relationnelle > Simple Query Format

L'algèbre relationnelle ne comprend pas d'opérations qui permettent de créer une relation, d'y introduire des données ou d'en afficher le contenu.

Ces opérations indispensables pour un projet automatisé sont définies dans SQF.

SQF comprend quatre instructions pour :

1. créer une relation :

```
@relation nom_relation:attribut1,attribut2,...attributZ
```

2. insérer des données dans une relation :

```
@data nom_relation:data1,data2,...dataZ
```

3. exécuter une opération de l'algèbre relationnelle :

```
@let nom_relation:<operation alg rel>
```

4. afficher le contenu d'une relation

```
@print nom_relation:
```

Critères de fonctionnement

- ◆ Facilité d'utilisation
- ◆ Syntaxe simple
- ◆ Indépendance d'autre logiciel
- ◆ Multi plateforme
- ◆ Code accessible et compréhensible

Fonctionnement > Fonctionnement attendu



Rép. perso



Démarrer ici



Corbeille



partage

```

data.sqf - KWrite
File Edit Bookmarks Tools Settings Help

@relation COURS:Nom,Prof,Ects
@data COURS:Gestion_de_projets,Mens,4
@data COURS:Bases_de_donnees,Wijsen,15
@data COURS:Analyse,Troestler,10
@data COURS:Data_Mining,Wijsen,10

@relation NOTES:Etud,Nom_cours>Note
@data NOTES:Ed,Gestion_de_projets,13
@data NOTES:Ed,Bases_de_donnees,15
@data NOTES:Tim,Bases_de_donnees,13
@data NOTES:Tim,Gestion_de_projets,19

@relation COURS2001:Nom,Prof,Ects
@data COURS2001:Gestion_de_projets,Bruyere,4
@data COURS2001:Bases_de_donnees,Wijsen,15
@data COURS2001:Analyse,Troestler,10
@data COURS2001:Data_Mining,Wijsen,6
@data COURS2001:OMT,GOOSSENS,10

Line: 18 Col: 31 INS NORM

```



[input.sqf - KWrite]	data.sqf - KWrite
[output2.sqf [modified] - KWrite]	[olivier@dell_oli:~/projet]

100% [Battery Icon] ? 11:16 ven 10 sep



Rép. perso



Démarrer ici



Corbeille



partage

```

data.sqf - KWrite
File Edit Bookmarks Tools Settings Help

@relation COURS:Nom,Prof,Ects
@data COURS:Gestion_de_projets,Mens,4
@data COURS:Bases_de_donnees,Wijsen,15
@data COURS:Analyse,Troestler,10
@data COURS:Data_Mining,Wijsen,10

@relation NOTES:Etud,Nom_cours>Note
@data NOTES:Ed,Gestion_de_projets,13
@data NOTES:Ed,Bases_de_donnees,15
@data NOTES:Tim,Bases_de_donnees,13
@data NOTES:Tim,Gestion_de_projets,19

@relation COURS2001:Nom,Prof,Ects
@data COURS2001:Gestion_de_projets,Bruyere,4
@data COURS2001:Bases_de_donnees,Wijsen,15
@data COURS2001:Analyse,Troestler,10
@data COURS2001:Data_Mining,Wijsen,6
@data COURS2001:OMT,GOOSSENS,10

Line: 18 Col: 31 INS NORM

```

```

input.sqf - KWrite
File Edit Bookmarks Tools Settings Help

@let CCOURS:COURS WHERE Prof=Mens
@print CCOURS:
@let PROF:COURS PROJECT Prof
@print PROF:
@let DETAILS:COURS JOIN NOTES
@print DETAILS:
@let COURS2:COURS RENAME Nom AS Nom_cours
@print COURS2:
@let C2000_2001:COURS UNION COURS2001
@print C2000_2001:
@let DIFF_C:COURS2001 MINUS COURS
@print DIFF_C:
@let COURS_WIJSEN:((COURS UNION COURS2001) WHERE Prof=Wijsen) PROJECT Nom,Ects
@print COURS_WIJSEN:

Line: 7 Col: 41 INS NORM

```



input.sqf - KWrite	data.sqf - KWrite
[output2.sqf [modified] - KWrite]	[olivier@dell_oli:~/projet]

System tray area containing icons for network, battery (100%), help, and system clock showing 11:16 ven 10 sep.



```

data.sqf - KWrite
File Edit Bookmarks Tools Settings Help
@relation COURS:Nom,Prof,Ects
@data COURS:Gestion_de_projets,Mens,4
@data COURS:Bases_de_donnees,Wijsen,15
@data COURS:Analyse,Troestler,10
@data COURS:Data_Mining,Wijsen,10

@relation NOTES:Etud,Nom_cours>Note
@data NOTES:Ed,Gestion_de_projets,13
@data NOTES:Ed,Bases_de_donnees,15
@data NOTES:Tim,Bases_de_donnees,13
@data NOTES:Tim,Gestion_de_projets,19

@relation COURS2001:Nom,Prof,Ects
@data COURS2001:Gestion_de_projets,Bruyere,4
@data COURS2001:Bases_de_donnees,Wijsen,15
@data COURS2001:Analyse,Troestler,10
@data COURS2001:Data_Mining,Wijsen,6
@data COURS2001:OMT,GOOSSENS,10
Line: 18 Col: 31 INS NORM

```

```

input.sqf - KWrite
File Edit Bookmarks Tools Settings Help
@let CCOURS:COURS WHERE Prof=Mens
@print CCOURS:
@let PROF:COURS PROJECT Prof
@print PROF:
@let DETAILS:COURS JOIN NOTES
@print DETAILS:
@let COURS2:COURS RENAME Nom AS Nom_cours
@print COURS2:
@let C2000_2001:COURS UNION COURS2001
@print C2000_2001:
@let DIFF_C:COURS2001 MINUS COURS
@print DIFF_C:
@let COURS_WIJSEN:(COURS UNION COURS2001) WHERE Prof=Wijsen) PROJECT Nom,Ects
@print COURS_WIJSEN:
Line: 7 Col: 41 INS NORM

```

```

olivier@dell_oli:~/projet
Fichier Éditer Paramètres Aide
[olivier@dell_oli projet]$ ./projet data.sqf input.sqf output2.sqf
SQF - Version Sep 2004 - www.christiaen.org/sqf
Dictionnaire : 13 relation(s)
NOM          - ADRESSE - COL - TUPL - T_ACT
COURS        134559640  3    5    4
NOTES        134565144  3    5    4
COURS2001    134570648  3    6    5
CCOURS       134576928  3    2    1
PROF         134582592  1    5    3
#DETAILS     134589240  6    17   16
DETAILS      134600928  6    17   16
COURS2       134606296  3    5    4
C2000_2001   134613040  3    10   7
DIFF_C       134620688  3    4    3
#J           134627096  3    10   7
#K           134634496  3    4    3
COURS_WIJSEN 134640488  2    4    3
[olivier@dell_oli projet]$

```



```

data.sqf - KWrite
File Edit Bookmarks Tools Settings Help
@relation COURS:Nom,Prof,Ects
@data COURS:Gestion_de_projets,Mens,4
@data COURS:Bases_de_donnees,Wijsen,15
@data COURS:Analyse,Troestler,10
@data COURS:Data_Mining,Wijsen,10

@relation NOTES:Etud,Nom_cours>Note
@data NOTES:Ed,Gestion_de_projets,13
@data NOTES:Ed,Bases_de_donnees,15
@data NOTES:Tim,Bases_de_donnees,13
@data NOTES:Tim,Gestion_de_projets,19

@relation COURS2001:Nom,Prof,Ects
@data COURS2001:Gestion_de_projets,Bruyere,4
@data COURS2001:Bases_de_donnees,Wijsen,15
@data COURS2001:Analyse,Troestler,10
@data COURS2001:Data_Mining,Wijsen,6
@data COURS2001:OMT,GOOSSENS,10
Line: 18 Col: 31 INS NORM

```

```

input.sqf - KWrite
File Edit Bookmarks Tools Settings Help
@let CCOURS:COURS WHERE Prof=Mens
@print CCOURS:
@let PROF:COURS PROJECT Prof
@print PROF:
@let DETAILS:COURS JOIN NOTES
@print DETAILS:
@let COURS2:COURS RENAME Nom AS Nom_cours
@print COURS2:
@let C2000_2001:COURS UNION COURS2001
@print C2000_2001:
@let DIFF_C:COURS2001 MINUS COURS
@print DIFF_C:
@let COURS_WIJSEN:((COURS UNION COURS2001) WHERE Prof=Wijsen) PROJECT Nom,Ects
@print COURS_WIJSEN:
Line: 7 Col: 41 INS NORM

```

```

olivier@dell_oli:~/projet
Fichier Éditer Paramètres Aide
[olivier@dell_oli projet]$ ./projet data.sqf input.sqf output2.sqf

SQF - Version Sep 2004 - www.christiaen.org/sqf

Dictionnaire : 13 relation(s)
NOM          - ADRESSE - COL - TUPL - T_ACT
COURS        134559640 3    5    4
NOTES        134565144 3    5    4
COURS2001    134570648 3    6    5
CCOURS       134576928 3    2    1
PROF         134582592 1    5    3
#DETAILS     134589240 6    17   16
DETAILS      134600928 6    17   16
COURS2       134606296 3    5    4
C2000_2001   134613040 3    10   7
DIFF_C       134620688 3    4    3
#J           134627096 3    10   7
#K           134634496 3    4    3
COURS_WIJSEN 134640488 2    4    3
[olivier@dell_oli projet]$

```

```

output2.sqf [modified] - KWrite
File Edit Bookmarks Tools Settings Help
@relation CCOURS:Nom,Prof,Ects
@data CCOURS:Gestion_de_projets,Mens,4

@relation PROF:Prof
@data PROF:Mens
@data PROF:Wijsen
@data PROF:Troestler

@relation DETAILS:Nom,Prof,Ects,Etud,Nom_cours>Note
@data DETAILS:Gestion_de_projets,Mens,4,Ed,Gestion_de_projets,13
@data DETAILS:Gestion_de_projets,Mens,4,Ed,Bases_de_donnees,15
@data DETAILS:Gestion_de_projets,Mens,4,Tim,Bases_de_donnees,13
@data DETAILS:Gestion_de_projets,Mens,4,Tim,Gestion_de_projets,19
@data DETAILS:Bases_de_donnees,Wijsen,15,Ed,Gestion_de_projets,13
@data DETAILS:Bases_de_donnees,Wijsen,15,Ed,Bases_de_donnees,15
@data DETAILS:Bases_de_donnees,Wijsen,15,Tim,Bases_de_donnees,13
@data DETAILS:Bases_de_donnees,Wijsen,15,Tim,Gestion_de_projets,19
@data DETAILS:Analyse,Troestler,10,Ed,Gestion_de_projets,13
@data DETAILS:Analyse,Troestler,10,Ed,Bases_de_donnees,15
@data DETAILS:Analyse,Troestler,10,Tim,Bases_de_donnees,13
@data DETAILS:Analyse,Troestler,10,Tim,Gestion_de_projets,19
@data DETAILS:Data_Mining,Wijsen,10,Ed,Gestion_de_projets,13
@data DETAILS:Data_Mining,Wijsen,10,Ed,Bases_de_donnees,15
@data DETAILS:Data_Mining,Wijsen,10,Tim,Bases_de_donnees,13
@data DETAILS:Data_Mining,Wijsen,10,Tim,Gestion_de_projets,19

@relation COURS2:Nom_cours,Prof,Ects
@data COURS2:Gestion_de_projets,Mens,4
@data COURS2:Bases_de_donnees,Wijsen,15
@data COURS2:Analyse,Troestler,10
@data COURS2:Data_Mining,Wijsen,10

@relation C2000_2001:Nom,Prof,Ects
@data C2000_2001:Gestion_de_projets,Mens,4
@data C2000_2001:Bases_de_donnees,Wijsen,15
@data C2000_2001:Analyse,Troestler,10
@data C2000_2001:Data_Mining,Wijsen,10
@data C2000_2001:Gestion_de_projets,Bruyere,4
@data C2000_2001:Data_Mining,Wijsen,6
@data C2000_2001:OMT,GOOSSENS,10
Line: 26 Col: 0 INS NORM *

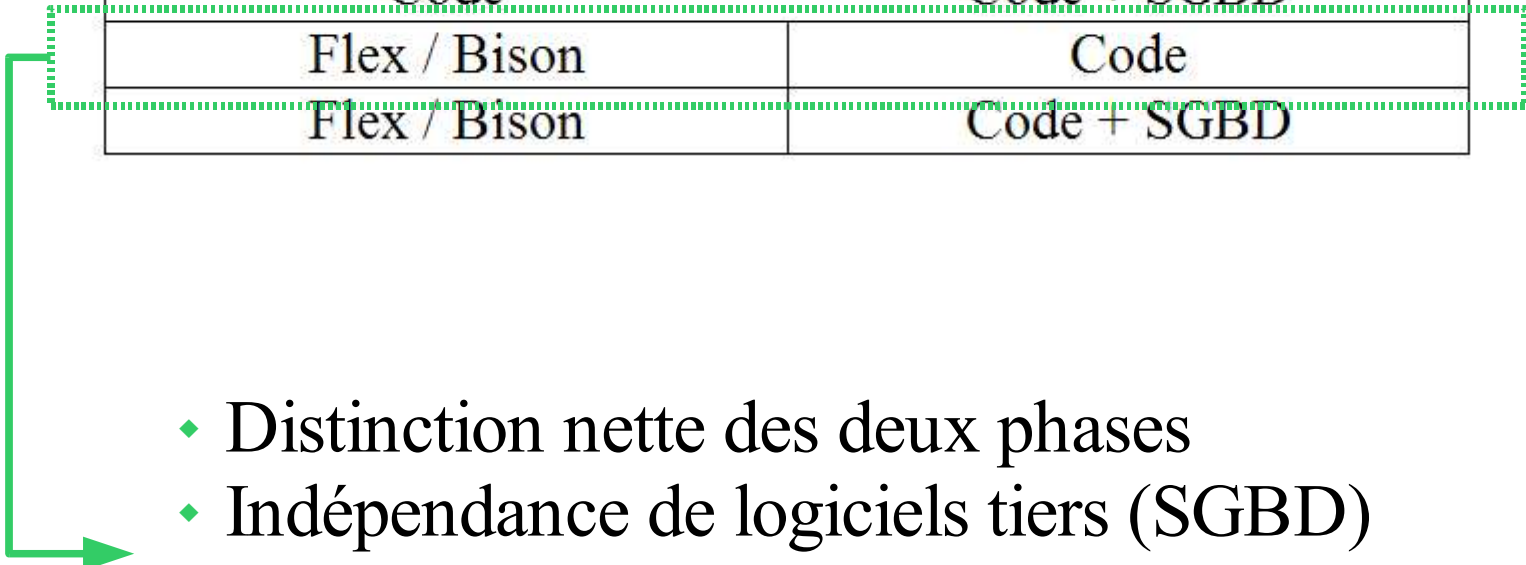
```

Développement > Approche

Analyse lexicale et syntaxique	Manipulation des données
Code	Code
Code	Code + SGBD
Flex / Bison	Code
Flex / Bison	Code + SGBD

Développement > Approche

Analyse lexicale et syntaxique	Manipulation des données
Code	Code
Code	Code + SGBD
Flex / Bison	Code
Flex / Bison	Code + SGBD



- ◆ Distinction nette des deux phases
- ◆ Indépendance de logiciels tiers (SGBD)
- ◆ Algorithmes d'analyse éprouvés
- ◆ Solution portable

1. Analyse lexicale et syntaxique des fichiers en entrée
2. Définition des structures de données
3. Ecriture du code assurant l'exécution des opérations de l'algèbre
4. Intégration de ces modules dans une seule application : l'interpréteur.



Le but de l'analyse lexicale est de **reconnaître** et de transformer une suite de symboles en terminaux.

Flex (Lex) est un utilitaire qui permet de générer automatiquement un analyseur lexical à partir d'une spécification basée sur un ensemble d'**expressions régulières** et produit en sortie le texte source d'un programme C.

Développement > extrait spécification Flex SQF

```
...
Chiffre      [0-9]
Lettre       [A-Za-z_]
Espace       [ ]
Mot          ({Lettre}+({Lettre}|{Chiffre})*)|{Chiffre}+
Liste_mots   ({Mot}[,])+{Mot}
Comparaison  [=<>]

%%
[ \t]        {}
[ \n]        {num_ligne++;}
"WHERE"      {return OP_WHERE;}
"PROJECT"    {return OP_PROJECT;}
...
"@relation " {return SQF_RELATION;}
"@data "     {return SQF_DATA;}
"@let "      {return SQF_LET;}
"@print"     {return SQF_PRINT;}

{Mot}        {
              yylval.val_char=(char *) malloc((1+yyleng)*sizeof(char));
              strcpy(yylval.val_char, yytext);
              return Mot;
            }
{Liste_mots}{
              yylval.val_char=(char *)malloc((1+yyleng)*sizeof(char));
              strcpy(yylval.val_char, yytext);
              return Liste_mots;
            }
...

```

Développement > Analyse syntaxique

L'objectif de cette analyse est de regrouper les terminaux fournis par l'analyse lexicale en phrases grammaticales

Définition de la syntaxe de SQF (formalisme EBNF):

```
<sequence_sqf> ::= <commande_sqf>+
<commande_sqf> ::= <sqf_relation> | <sqf_data> | <sqf_let> | <sqf_print>
<sqf_relation> ::= @relation <mot> : <tuple>
<sqf_data>     ::= @data <mot> : <tuple>
<sqf_print>    ::= @print <mot> :
<sqf_let>     ::= @let <mot> : <relational operation>
<relational expression> ::= <nom relation> | <relational operation> | (<relational expression>)
<relational operation> ::= <select> | <project> | <join> | <rename> | <union> | <difference>
<select>       ::= <relational expression> WHERE <mot> <comparaison> <mot>
<project>     ::= <relational expression> PROJECT <tuple>
<join>        ::= <relational expression> JOIN <relational expression>
<rename>      ::= <relational expression> RENAME <mot> AS <mot>
<union>       ::= <relational expression> UNION <relational expression>
<difference>  ::= <relational expression> MINUS <relational expression>
<tuple>       ::= <mot> | <liste_mots>
<mot>         ::= <lettre>+(<lettre>|<chiffre>)* | <chiffre>+
<liste_mots>  ::= (<mot>,)+ <mot>
<comparaison> ::= = | < | >
```

Développement > Structures des données

La relation

nom - nombre de colonnes - nombre de tuples - nombre de tuples actifs - le nombre maximum de tuples - un tableau de pointeur sur des tuples

Le tuple

état - longueur du tuple - un tableau de chaînes de caractères

Le dictionnaire des relations

nombre de relations existantes

tableau de pointeur sur les entrées du dictionnaire :

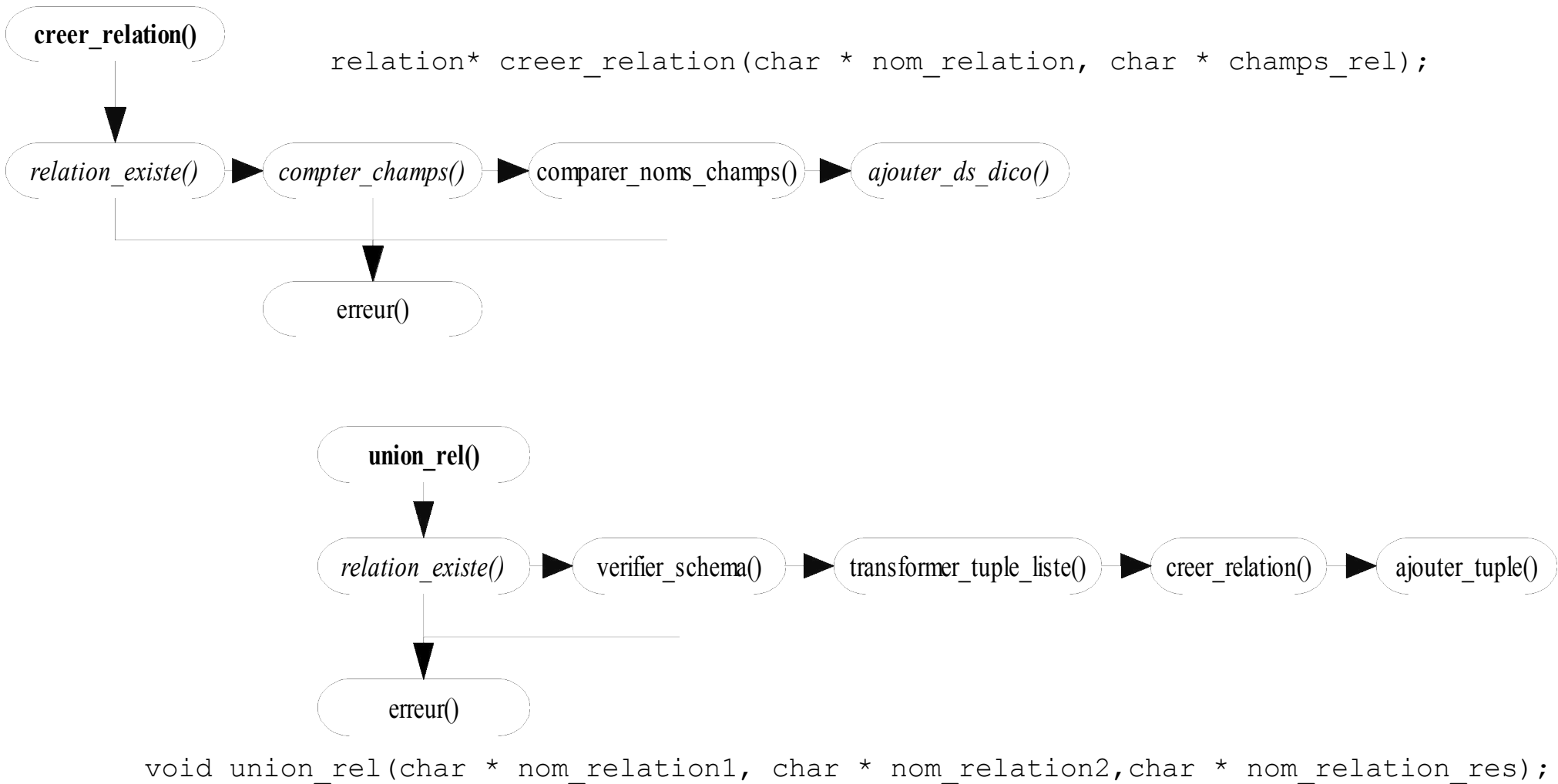
nom de la relation

adresse de la relation

Toute manipulation de relation(s) nécessite l'accès au dictionnaire.

Développement > Code d'exécution des opérations

Les fonctions principales recourent à des fonctions auxiliaires. >>
Chaque opération de l'algèbre est implémentée dans une fonction.

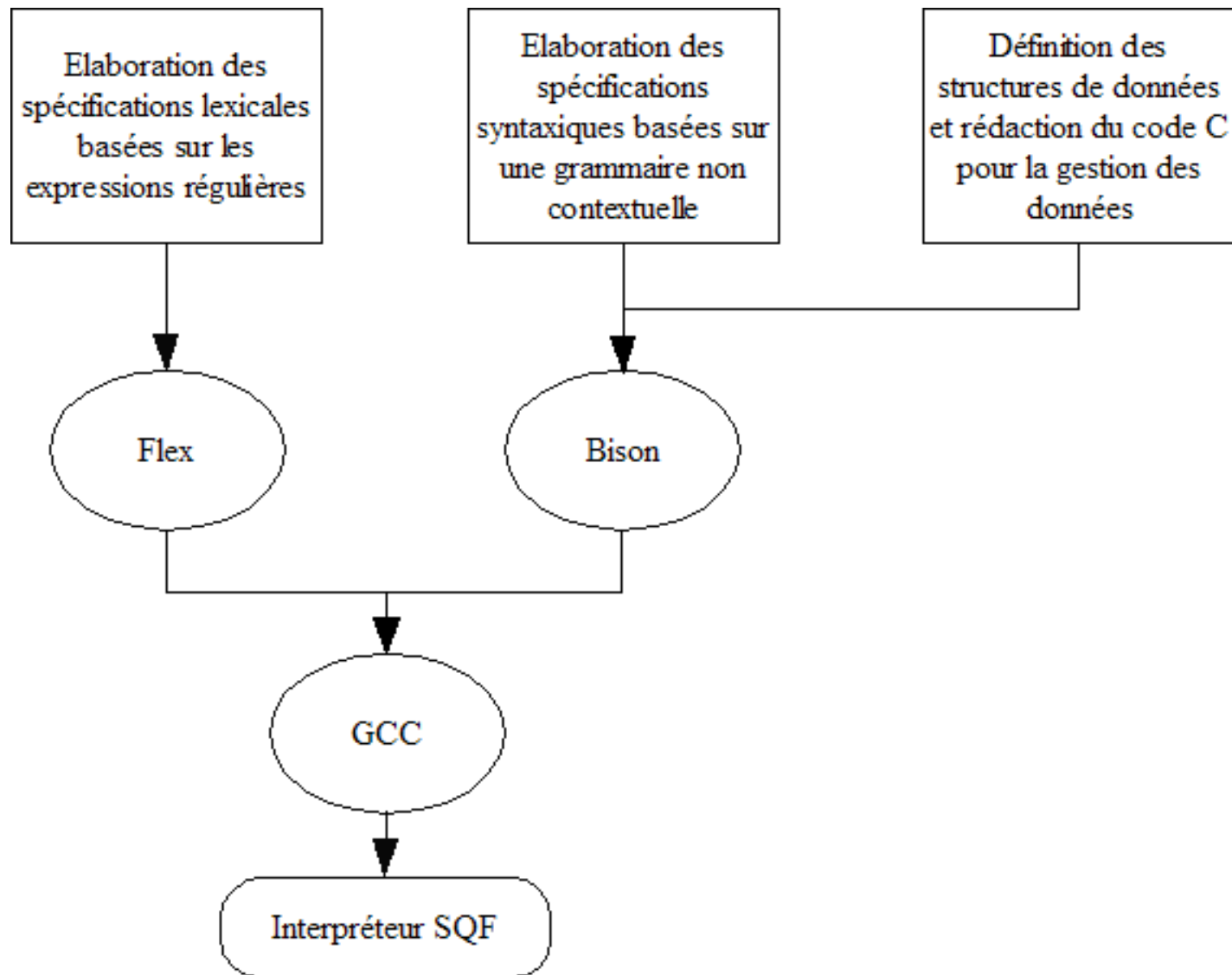


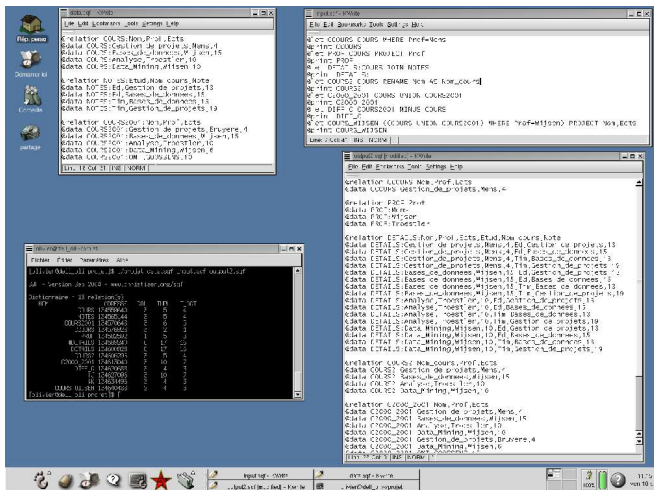
Développement > extrait spécification Bison SQF

Bison utilise des règles inspirées du formalisme BNF et permet d'y associer une action.

```
...
sequence_sqf  : commande_sqf
               | sequence_sqf commande_sqf
;
commande_sqf  : sqf_relation
               | sqf_data
               | sqf_let
               | sqf_print
;
sqf_relation  : SQF_RELATION Mot ':' tuple {creer_relation($2,$4);}
;
sqf_data      : SQF_DATA Mot ':' tuple      {ajouter_tuple($2,$4);}
;
sqf_let       : SQF_LET Mot ':' relational_operation {executer_operation($2,$4);free($4);}
;
...
relational_expression : Mot                {$$ = garnir_struct(0,$1,"","","");}
                      | relational_operation {executer_operation($1->nom_resultante,$1);}
                      | '('relational_expression' ) {$$=$2;strcpy($$->nom_resultante,$2->nom_resultante);}
;
...
r_union       : relational_expression OP_UNION relational_expression
               {$$ = garnir_struct(5,$1->nom_resultante,$3->nom_resultante,"","");}
;
tuple        : Mot      {$$=$1;}
              | Liste_mots {$$=$1;}
;
...
```

Développement > Intégration





- ◆ Concrétisation du projet
- ◆ Correspond aux critères initiaux
- ◆ Travail de recherche : étude préliminaire
- ◆ Importance de la méthodologie
- ◆ Développement de nouvelles compétences
- ◆ Logiciels libres

www.christiaen.org/sqf