Course support

INFORMATION SYSTEM & DATA BASE

Produced by

Narcisse TALLA

Department of Computer Science - IUT Fotso Victor of Bandjoun

http://www.lets.africa-web.org

ntalla@gmail.com

2008-2009

SOMAIRE

Chapter I. THE INFORMATION SYSTEM

1.1. Some definitions

❖ An information is the result of a process of formatting and materialization to communicate a fact or set of facts to an audience.

❖ A system is a set of dynamically interacting elements organized according to a goal.

❖ An **open system** is in permanent interaction with its environment: It is a reservoir that fills up (energy, matter, information) and empties (entropy, energy waste) at the same speed to stay within a given state. It is in constant interaction with its environment, modifying it and in return being modified.

❖ A **closed system** doesn't exchange nor matter, nor information, nor energy: it lives on its internal energy and as feedback, increases its entropy which becomes maximum.

❖ A **complex system** consists of a wide variety of components or parts with specialized functions:

- Elements organized in **hierarchical levels**,
- Elements and levels connected by multiple **links**,
- **Non linear** interactions,
- Difficult and **unpredictable** behavior
- **Great resistance** to change.

Any system has two aspects:

❖ The structural aspect (structural or spatial description).

❖ The functional aspect (temporal description).

1.2. The concept of "information system".

The concept of information system is a product of the general systems theory, developed in the 1940s by Van Bertalanffy and developed in the 1950s, notably in France by Jean-Louis Le Moigne. What characterizes the systemic approach can be seen as a reaction to an analytical approach typical of the Cartesian logic and it should be noted that systemic stresses on the concept of totality.

For example, According to Saussure, a system is an "organised whole, made up of elements that can only be described relative to each other according to their place in this totality". According to Von Bertalanffy, it is a "set of units in mutual interrelations". Lesourne to his side talk of a "set of elements linked by a set of relations."

The systemic concept distinguishes systems and subsystems. For example in a car, the engine will be a subsystem. In the human body, the digestive system is a subsystem.

1.3. The systemic

To examine the infinitely small and the infinitely large, scientists use tools like the microscope and the telescope that have led to significant leaps of knowledge. Today, we are confronted with the **infinitely complex** and there are no tools to study the tremendous complexity of the systems we are elements and particles. The systemic approach aims to develop a symbolic instrument that Joël de Rosnay calls **macroscope** to study this complexity. However, the systemic approach does not pretend to explain everything or to solve everything and is not intended to provide models of the world that claim to encompass everything.

The natural tendency of the human spirit is unifying, reductive, approximating; It gives apparently satisfactory model, but dangerous: one filters and eliminates all what is different from the unified model and come out with the worst intransigence. The systemic models are starting points for reflection; they must be confronted with reality, they must be attacked, destroyed, to be better rebuilt because they can **evolve** only in **confrontation**.

1.4. Some methods of analysis

In computer science analysis, there are several methods including:

1.4.1. The method FAST (Function Analysis Systems Techniques)

General principle: graphic communication instrument among stakeholders, FAST focuses on the relationships between products and functions. This instrument allows representing the logical relationships between tasks by repetition of "why? / How? / When?" posed at each stage of the analysis. FAST presents the relationship between products and functions within a strictly delimited domain.

Originalities: FAST produces a diagram that allows the designer to explain and justify the technical solutions.

1.4.2. The method APTE (APplication des Techniques d'Entreprise)

General principle: A set of methodological tools available to project teams to control the consistency between the various elements of a project, to make performed and reliable technology choices to satisfy the needs at lower cost. This method provides common logics of formalisation for conceptual phases of complex projects.

Originalities: Very systemic in the sense that the product has been designed from its surrounding environment, favouring the point of view of the designer and the user. APTE has its own internal validation tools (need, functions, constraints...). APTE offers an effective method of group animation.

1.4.3. The method GRAFCET

General principle: GRAFCET proposes a graphical representation of awaited behaviour of a logical system. "Make a GRAFCET" consists of describing sequential ongoing starting from functional specifications of an automated equipment. A GRAFCET allows to describing the "command" part of the automated system.

Originalities: GRAFCET is used only for automatism. The GRAFCET language is used to program programmable automates.

1.4.4. The method RELIASEP

General principle: From the need expressed in term of functions, find the sub-functions required to satisfy the need (functional tree). This research is progressive and adapted to the development phases.

Originalities: Very oriented operating security, its functional tree is used as basis to analyse failure modes, their effects and their criticality (FMECA). This method allows visualising the functional path of deterioration. RELIASEP is effective for the study of complex systems more oriented "material" than "software".

1.4.5. The method SA/RT

General principle: From an analysis based on relevant questions and the graphical representation of this analysis, this method allows the development of:

- A model of the analysed system
- A model of the static process attached to it
- A dynamic control model that will help for the method use

Originalities: This method considers the dynamic aspect of the analysed system. It is based on an old method (Structured Analysis -SA) that had been widely used.

- SA / RT is well suited for applications with high dynamic behaviour.
- SA / RT is rather focused on sharing the work and on human-machine interface.

1.4.6. The SADT method

Definition : The acronym SADT means System Analysis and Design Technic. This method has been developed by the Softech company in the United States. SADT is a method of analysis by successive levels of descriptive approach of a given set.

General principle: Modelling of existing or future systems to understand the functioning and to envisage adequate solutions. This modelling is based on the actions of the analysed system (actigram) and on the data the system can manage (datagram) in an arborescent structure of the system.

Originalities : Rather oriented information system, softwares and automatisms, SADT defines two types of stakeholders: the author who conceives the system and the reader who criticises. These works allow to precise and to optimise the system by iterative approach.

SADT allows a good traceability and internal controls of the coherence. It can be summarised as:

- A graphical tool of representation
- It obliges to record the decisions of the work team. This progressively allows to create a complete documentation of the system.
- It's a team effort that requires discipline and coordination. Indeed, SADT is a product for communication and diffusion.
- Its formalism leads to and ascendant or descendant structured representation.

- If SADT is completely used (actigram and datagram), it allows to program directly an automated system.

1.4.7 The method MERISE

General principle: Representation of and existing system by easiest invariant elements (sub-systems of order n) which compose the system, optimisation of links and relations required for the functioning of the system.

Originalities: MERISE is very oriented information system and software. Rather old, it's progressively enriched until the appearance of "clones of MERISE".

Chapter II. Study of a modelling method : MERISE

2.1. Introduction

The conception of an information system is not obvious. Indeed, one must think about the whole organisation to be put in place. The conception phase requires methods allowing building the basic model. The modelling consists of creating a virtual representation of a reality in order to highlight the interesting points. This kind of method is called *analysis*. There are several methods of analysis and MERISE is one of the most used.

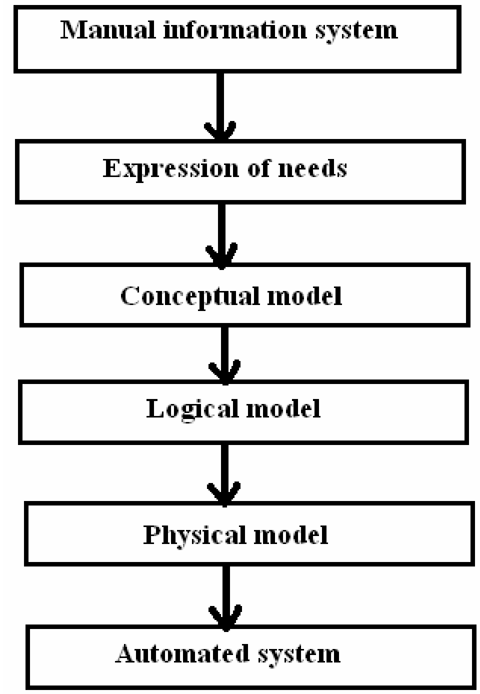
2.2. Presentation of MERISE

2.2.1. Definition and generalities

MERISE is a method of conception, development and realisation of automated systems. The purpose of this method is to conceive an information system. MERISE is based on the separation of data from treatments to be done in several conceptual and physical models.
The separation of data from operations insures longevity to the model. Indeed, the arrangement of data doesn't often need to be reshuffled, while operations or treatments are being frequently.
Regarding the company, MERISE distinguishes 3 sub-systems:
Steering system
Information System
Operating system
Where operating systems concerns basic operations of the enterprise (production of goods for sale) and the steering system points to the organisation and decision instances. Between the two, MERISE situates the information system.

2.2.2. Abstraction cycle of Information System conception

The conception of information system is done by steps, in order to lead to a functional information system reflecting a physical reality. The purpose is to validate one by one each step, considering the results of the previous steps. Since data are separated form operations, it is required to verify that all data required by the system are present and that there is no superfluous data. This succession of steps is called "abstraction cycle" for the conception of information system:



a- The Manual information system

The manual information system is a set of manuscript and oral information used for the management of the structure. This information is generally presented in the form of forms to fill or in the form of report to interpret. In the analysis process, this step consists in realising interviews beside different services managers or beside system managers and in collecting samples of each type of information handled in the system.

b- The Expression of needs

This step consists in defining what is expected from the automated information system. Therefore, it is required to:

- Do the inventory of required elements for the information system,
- Identify the limits or difficulties of the system management,
- Delimit the system by getting information beside the future users.

c- The Conceptual model

This step consists in realising different conceptual models among which the fundamentals are: the Conceptual Model of Data (MCD) and the Operating Conceptual Model (MCT)

The MCD is the model on which are based the data of the system. It describes the architecture of the data base and presents the rules and constraints to consider.

After the expression of needs, the specification of the Conceptual Model of Communication (MCC) which defines the information flows to consider can be done. These information flows start from and actor to another actor of the system and specify the nature or the type and the orientation of the transmitted information. The Operating Organisational Model (MOT) describes the constraints due to the environment (organisational, spatial and temporal).

d- The Logical Model

The Logical Model represents a software choice for the information system. It is deduced from the MCD and help in writing queries to access data bases. It presents the data base in the relational form and expresses the associated functional dependences.

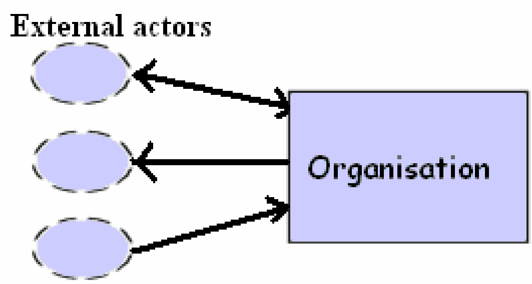
e- The Physical Model

It reflects the matter choice (for example, Ms Access, Oracle, Paradox, Mysql, etc.) for the information system. It presents the organisation of the created tables and the different relations between them.

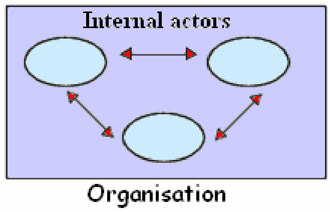
2.3. The Conceptual Model of Communication (MCC)

2.3.1. Definition of the organisation

- The firts step of this model is to isolate the system by delimiting it. The purpose is to define the system and the external elements which exchange information flows with the system. These external elements are called external stakeholders/actors or partners. Information exchanges can be in only one direction (from external actor to the system exclusively or from the system to the external actor exclusively) or in double direction (simultaneously from the external actor the system and vice versa).



- The second step consists in cutting up the organisation into entities called internal stakeholders/actor (or domaine). When the domains of an organisation are very large, the can be cut up again into sub-domains.

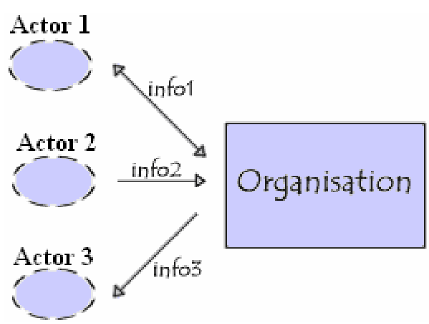


- The last step is the analysis of the information flows, that is, the definition of process.

2.3.2. The Context Diagram

The context diagram aims to represent the information flows between the organisation and the external actors accordingly to a standardised representation in which each object has a name. Notably:

- The organisation is represented by a rectangle,
- External actors are represented by ellipses in dotted line,
- The information flow is represented by an arrow which orientation specifies the direction of the information flow.

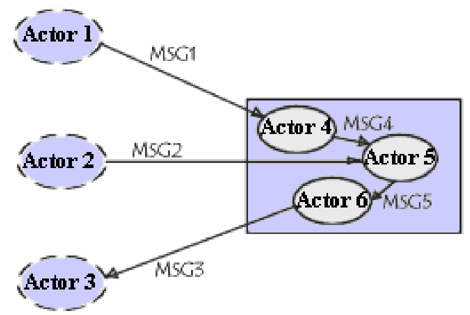


2.3.3. Conceptual Diagram of Flow

This diagram (also called Conceptual Model of Communication) helps to complete the context diagram by breaking down the organisation into a series of internal actors. In this diagram, the standard representation is as follow:

- Internal actors are represented by ellipses in continuous line,

- Internal messages are represented by arrows



2.4. The Conceptual Model of Data (MCD)

2.4.1. The purpose

The MCD aims to describe formally the data to be used in the information system. It is question here of a data representation, easily comprehensible, allowing describing the information system using entities.

2.4.2. Entities and class of entities

An entity is the representation of a material or immaterial element having a role in the information system that one whish to describe.

A class of entities is a set of entities of the same type, (having the same definition). An entity is an instantiation of the class. Each entity is composed of properties, elementary data describing the entity.

Let us consider the example of a Toyota KE70, a Relault 12 and a Peugeot 504. These are three entities belonging to the class "*Car*". The Toyota KE70 is then an instantiation of the class "*Car*". Each entity can possess the properties colour, year and model.

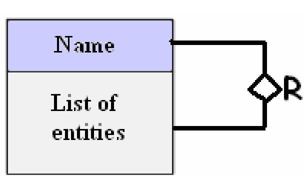
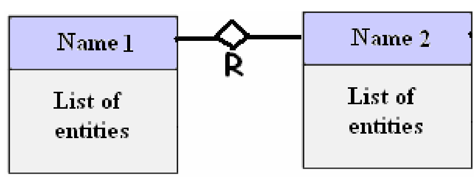
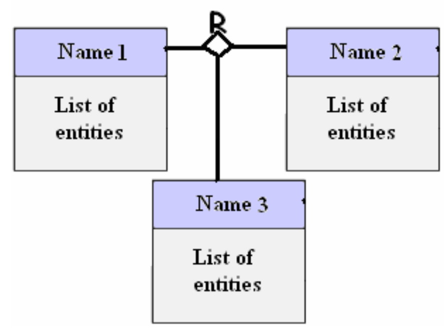
A class of entities is represented by a rectangle separated into two fields:

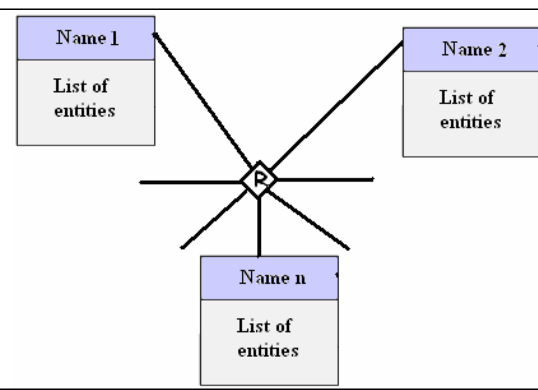
The upper field contains the name of the class. This name is generally an abbreviation for a matter of writing simplification. The question here is to verify that, to each class of entities corresponds one and only one name and vice versa.

The lower field contains the list of properties.

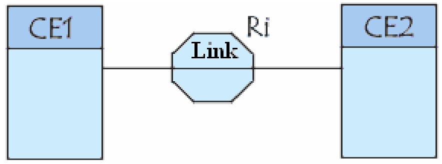
2.4.3. Relations and class of relation

A *relation* (also called *association*) represents the semantic links that can exist between several entities. A *class of relation* contains relations of the same type (It links entities belonging to the same class of entities). A class of relation can link more than two classes of entities. According to the number of participants in a relation, we have the following denominations:

<table>
<tr>
<td>
</td>
<td>
</td>
<td>
</td>
</tr>
<tr>
<td>A recursive (reflexive) class of relation links a class of entity to itself.</td>
<td>A binary class of relation links two classes of entity</td>
<td>A ternary class of relation links three classes of entity</td>
</tr>
</table>

<table>
<tr>
<td></td>
<td>
</td>
<td></td>
</tr>
<tr>
<td></td>
<td>An n-ary class of relation links n classes of entity.</td>
<td></td>
</tr>
</table>

Classes of relation are represented by *hexagons* (at times by *ellipses*). The name (generally a verb) describes the type of the relation that links the classes. For each class of relation, an identifier *Ri* is defined. This identifier defines one and only one class of relation.

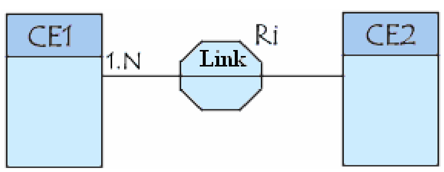


One can eventually add properties to classes of relation. These properties are called *association properties*. These properties can not belong to any class of relation taking part to the association. For example, the mark is an association property between the class of entity *Course* and the class of entity *student*.

2.4.4. The cardinality

Cardinalities characterise the link between an entity and an associated relation. The cardinality of a relation is a couple made up of a minimum limit and a maximum limit defining the interval in which the cardinality of an entity can take its value:

- The minimal limit (generally 0 or 1) describes the minimum number of times that an entity can participate to the associated relation.
- The maximal limit (generally 1 or n) describes the maximum number of times that an entity can participate to the associated relation.



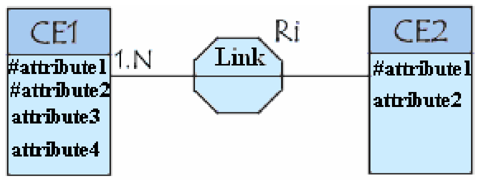
A cardinality 1.N means that each entity belonging to a class of entity participates at least one time to the associated relation (link).

A cardinality 0.N means that each entity belonging to a class of entity does not absolutely participate to the associated relation (link).

2.4.5. Identifiers

An identifier is a set of properties (one or several) describing one and only one entity. The original definition is the following:

The identifier is a particular property (set of properties) of an object so that it's impossible to find two occurrences of the said object for which this (these) property(ies) should have the same value.
Attributes of a class of entity identifying one and only one entity are called *absolute identifiers*. The MCD formalism proposes to precede identifiers with # or to underline them.



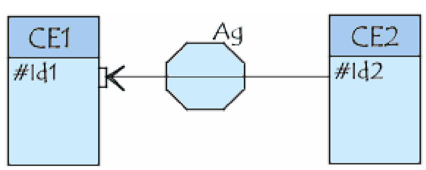
Then, each class of entity must own at least one identifier attribute and the whole set of its identifiers attributes must be field during the creation of the entity.

2.4.6. Aggregation (or relative identification)

When an entity identifier is constitute only of intrinsic attributes (attributes that doesn't refer to another entity), it is called *absolute identifier*. Entities made up of absolute identifiers can be defined independently to other entity occurrences. In this case, entities are *independent*.
However, some entities can only be identified through others. This is why we talk of relative identifier. For example, one can talk of the 4th door of the 2nd floor of the building B instead of door N°3451…
Thus, an aggregation specifies that an entity is required for the identification of another one.

- The class of entity required for the identification of another is called the aggregator class.
- The class identified is called the aggregated class.

The representation of this type of association is as follow:



2.5. The Operationg Conceptual Model (MCT)

2.5.1. Introduction

The operating conceptual model allows studying the dynamic of the information system.

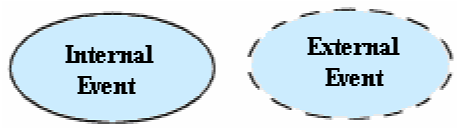
This model allows representing graphically the activity of the information system without considering organisational choices or execution means. It defines simply what is supposed to be done without specifying neither the date, nor how, nor the place.

2.5.2. The concept of event

An event represents a change in the external universe of the Information system or in the information system itself.

- An external event is a change of the external universe of the information system
- An internal event is a change in the information system

Events are represented by ellipses in continuous line for internal event and dotted line for external event.



2.5.3. Process

A process is a sub-set of the entreprise activity. Indeed, the activity of the enterprise is made up of a set of process. A process itself is made up of *operations*.

2.5.7. Operation

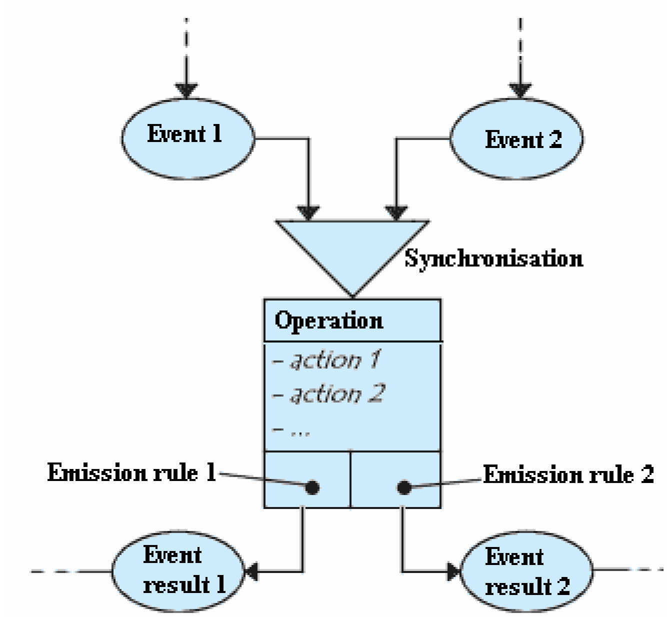
An operation is a set of actions proceeded by the system after an event or a conjunction of events. This set of actions is uninterruptible until the operation is accomplished.

2.5.8. The synchronisation

The synchronisation of an operation defines a Boolean condition on events that contribute to trigger the operation.

2.5.9. The construction of the operating conceptual model (MCT)

The operating conceptual model allows representing graphically the management of events.



2.6. The Operating Organisational Model (MOT)

2.6.1. Description

The operating organisational model describes the properties of operations not considered by the conceptual model of data, notably the time, resources and the place.
The MOT consists in representing the MCT in a table which specifies in its columns the duration, the place, the personal chair and the resources required by an action.

2.6.2. The table of functional procedures

The first step of the MOT consists in cutting up operations into functional procedures (FP), a succession of operations triggered by an event. The purpose is to associate in a table:

- The functional procedures,
- The beginning and the ending time,
- The place of the workstation,
- The personal chair of the workstation,
- The resources of the workstation

Procedure	Time		Workstation		
	Begin	End	Place	Personal chair	Resources

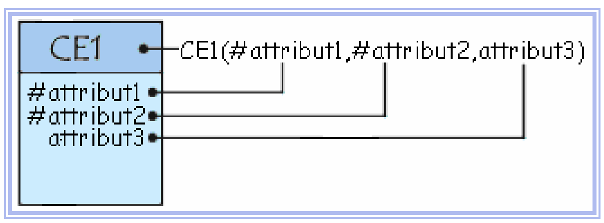
2.7. The logical model of data (MLD)

2.7.1. Description

The logical model of data consists in describing the structure of data used, without however referring to a particular programming language. The purpose is then to precise the type of data used during the process. Thus the logical model depends to the type of data base used.

2.7.2. Translation of a class of entity

Each class of entity of the conceptual model becomes a table in the logical model. The identifiers of the class of entity are called keys of the table while the standard properties become attributes of the table. Each attribute defines a column in the table.

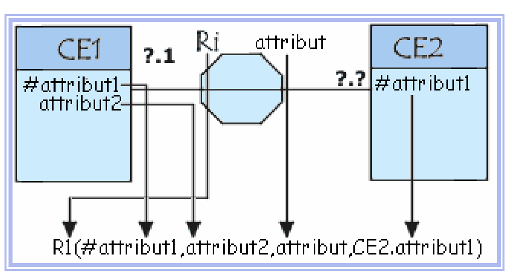


2.7.3. Translation of a class of relation

The passage from the conceptual model to the logical model related to the classes of relation is done accordingly to the cardinalities of each entity associated to the association.

If a class of entity possesses a weak cardinality, the table should have as attributes the properties of the class with weak cardinality, the properties of the relation and finally the primary key of the other class.

If the two classes of entity possess high cardinalities, the table should have as attributes, the primary keys of the two classes and the properties of the association.



2.7.4. Translation of an aggregation class

In the case of an aggregation class, the aggregated entity class has as additional attribute, the primary key of the aggregator class.

2.8. The physical model of data

This step consists in implementing the model in a specific System of data Base Management. This means to translate into a specific definition language. The commonly used language for this type of operation is the SQL (Structured Query Language).

2.9. Exercises

2.9.1. Automated management of a library

You have to create a data base for a library. The librarian asks you to be able to manage the following information: Books characterised each by a title, the authors (name and surname), an editor, an edition date and the kinds (for example: love, fiction science, politic, roman, news,…) and the customers (name and surname) each with an address (street, postal code and town), the books borrowed, the expected date for book return and the effective date of book return.
You don't know more and you are asked to do reasonable hypothesis and choices required by the system. Conceive then the data base.

2.9.2. The management of a urban transport company

You must create a data base for a bus company. The company manager gives you the following indications: each bus is numbered. It realises ways made of a departure, several stops and a terminus. Each bus can every day be allocated to several ways and it can do many journeys on each way with eventually distinct drivers. The management of customers is done separately (in another data base not concerned for this project) but the management of buses (date of the next revision, the mileage, number and name of the depot,…) and drivers (name, surname, grade, seniority, salary, buses droved, ways done,…) is included in this data base.

Chapter III. MICROSOFT ACCESS ENVIRONMENT

3.1. Ms Access home page

At the lunching of Ms Access, the following window appears.

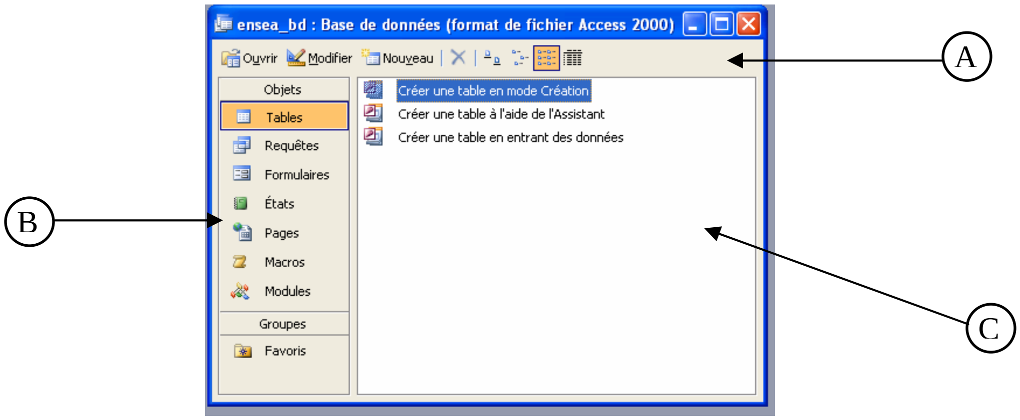


Figure 1: Home page

This window is divided into three blocs:

1. The menu bar (A)
2. The column of objects (B)
3. The application frame (C)

3.2. The menue bar (A)

The menu bar includes 8 menus subdivided into 3 categories: The first consists in creating, modifying and opening of objects (table, form, query,…) It contains the submenus Open, Modify and Create. The second category consists in deleting objects. The last category consists in presenting different objects in the application frame (AF).

3.2.1 *The submenu «Open»*

This option is active only when at least one object is selected in the AF. It allows opening for previewing selected objects in the AF.

3.2.2 *The submenu « Modify »*

The submenu Modify works typically as the submenu open. The unique difference is that objects are open for modification of their structures.

3.2.3 The submenu «New»

The submenu “New” allows creating samples of selected objects in column B. The default selected object is “Table”.

3.2.4 The submenu “delete”

The submenu “delete” allows deleting selected objects in the AF.

3.2.5 The submenu “Great icons”

The submenu “Great Icons” allows displaying objects icons in great format in the AF.

3.2.6 The submenu “small icons”

This submenu allows displaying objects icons in miniature format in the AF.

3.2.7 The submenu “List”

This submenu allows displaying objects’ icons in columns in the AF.

3.2.8 The submenu “Details”

This submenu allows displaying simultaniously icons and properties (name, description, type, date of creation and the date of the last modification) of every object in the AF.

3.3. The column of objects (B)

This column contains all objects handled by the data base.

3.3.1 Tables

A tables stores information of the data base. Indeed, the data base is a collection of tables in interaction each other.

3.3.2 Queries

Queries allow accessing the data base. Through queries, one can extract, insert or update data.

3.3.3 Forms

Forms are data base mining interfaces. They are intuitive, simple and convivial for the users hiding thus the operating complexity of the data base.

3.3.4 states

States allow presenting situations in the evolution of the data base. A state allows formatting a view of the data base by including whished information such that the view can be printed on an ordinary paper.

3.3.5 Pages

Pages allow defining data base access interface via the web.

3.3.6 Macros

Macros allow combining integrated functions in the data base management system to execute some operations.

3.3.7 Modules

Modules are used to insert a program (VBA: Visual Basic for Access) that can be called from the data base administration.

3.4. The application frame (C)

The application frame displays objects' icons related to the selected objects in the object column. It also displays the methods related to the selected objects in the object column. Generally, it proposes 2 basic functions for the creation of an object:

- Create an object in creation mode: this option is indicated for experts in the domain.
- Create an object in computer assisted mode: This option is recommended for amateurs in the domain.

Chapter IV. EXAMPLE OF DATA BASE CREATION

4.1 Problem

4.1.1 Specification

Let consider the case of realisation of the data base of the students of the national higher school of applied statistics and economy (ENSEA) of Abidjan in Côte d'Ivoire. In this data base, we are called to manage notably: (1) students characterised each by a registration number, a name, a surname, the year of first registration, the date of bird and the sex; (2) courses characterised each by au code, a name, a level and the hourly quota; (3) marks of students, teachers characterised each by a code, a name, a surname, a speciality, a grade and a sex; (4) the option (Mathematic, Economy,…), the category (ISE, ITS,…), the level and the function (delegate or not) of each student; The constraints are as follow.

1. The registration code belongs to one and only one student.
2. The registration code is sufficient to determine all other characteristics of the student.
3. The mark depends on the student, the course and the year.
4. Each course has only one principal teacher.
5. The same teacher can be principal teacher for many courses, independently to the level of the course.

4.1.2 The expected result.

The data base should produce answers to the following preoccupations:

- Extract all female students who have a mark greater than 12 in Information System in 2009.
- Extract all the principal teachers for the course Data base since 2007.
- Extract and order by level and by option all the delegates.

4.2 Solution

4.2.1. Analysis

One can denote that there are three entities, notably: student, teacher and course. The related conceptual model is the following:

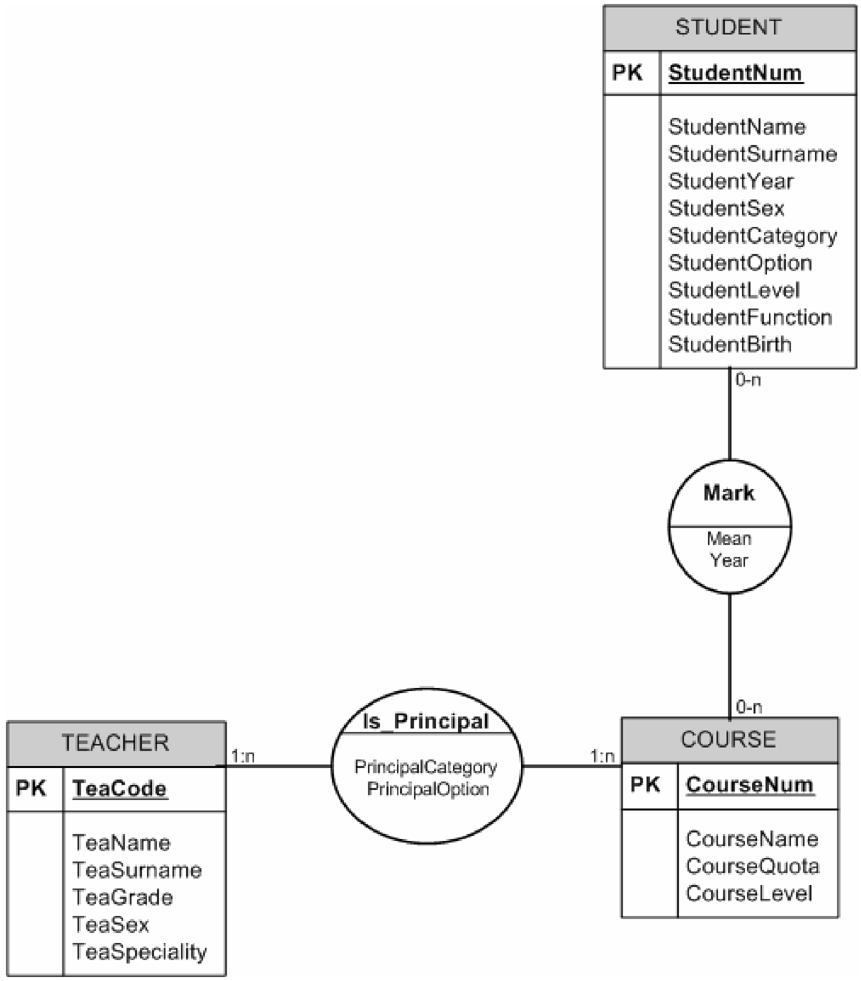


Figure 2: The conceptual model of data (MCD)

The equivalent logical model of data is the following:

```
Student (StudentNum, StudentName, StudentSurname, StudentYear,
StudentSex, StudentCategory, StudentOption,
StudentLevel,StudentFunction, StudentBirth)
Teacher (TeaCode, TeaName, TeaSurname, TeaGrade,TeaSex,
TeaSpeciality)
Cours (CourseNum, CourseName, CourseQuota, CourseLevel)
Mark (StudentNum,CourseNum, Mean,Year)
Principal (TeaCode, CourseNum, PrincipalCategory,PrincipalOption)
```

4.2.2 Creation of tables

One must insure that the object “table” is selected in the column of objects before double-clicking on the function “Create a table in creation mode” in the AF. The following interface is obtained.

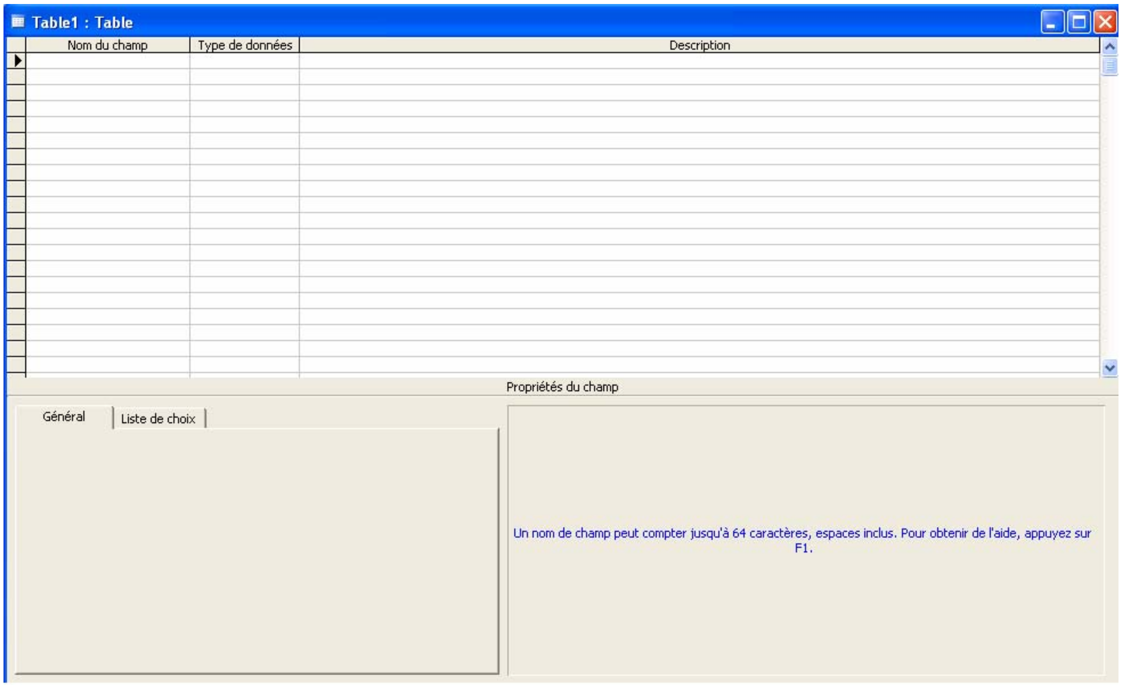


Figure 3: *Table creation form*

The column "Field name" contains the attributes of the table. The column "Type of data" contains the type (numerical, text…) of each attribute.

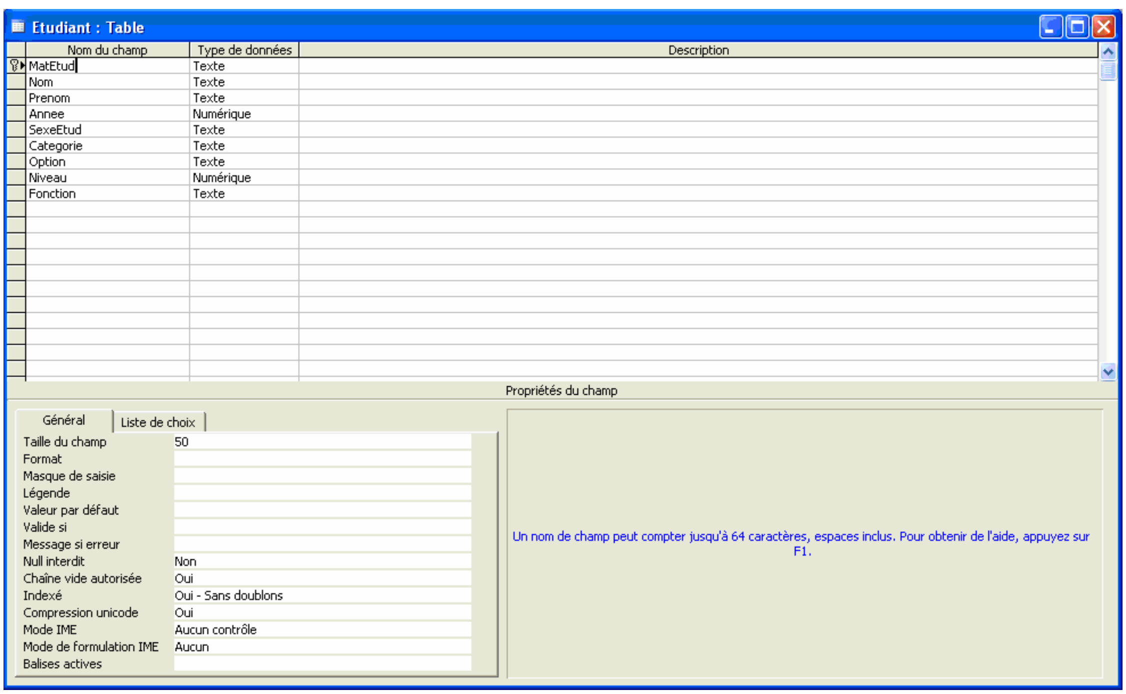


Figure 4: *Creation of the table «student»*

The window of the data base (figure 1) is presented in the following form:

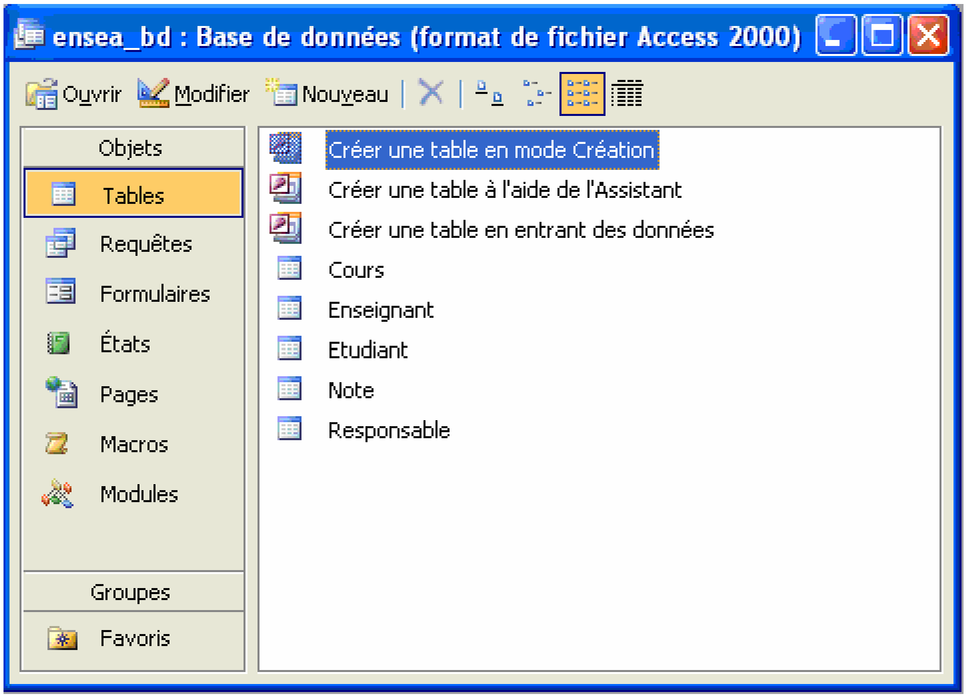


Figure 5: The tables of the data base ENSEA_BD

4.2.3. Creation of the physical model

After creation of tables, one must specify the relations existing among them in accordance with the MCD. One must therefore click on the submenu "relation…" of the menu "tools" in the bar menu to display the following page.

Figure 6: Entities of the data base

One must double-click on each table to integrate it in the model. From integrated tables, one constructs the physical graphic of the data base (figure 7). To materialise the type "one to many (1-n)" of the relation, One must click and maintain the mouse pressed on the attribute of the primary table (side of "1"), then displace and release on the corresponding attribute in the secondary table (side of "n"). The constraint is that the two attributes must be of the same type and the same length.

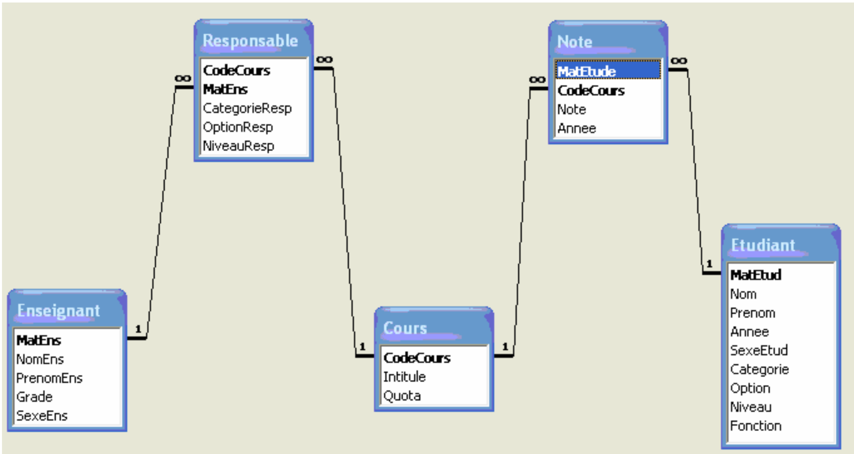


Figure 7: Physical model of data

4.2.4 Creation of queries

4.2.4.1 The QBE (Query By Exemple)

QBE is an interface of queries creation. Its limits are the facts that it does not support complex queries. To lunch it, one must select the object "Query" in the objects column to display the following page.

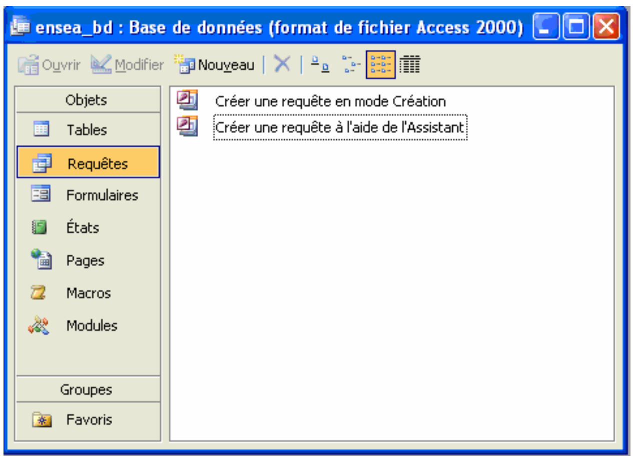


Figure 8: Window of creation of queries

From the previous window in figure 8, one must choice the function “Create a query in creation mode” to display the following page.

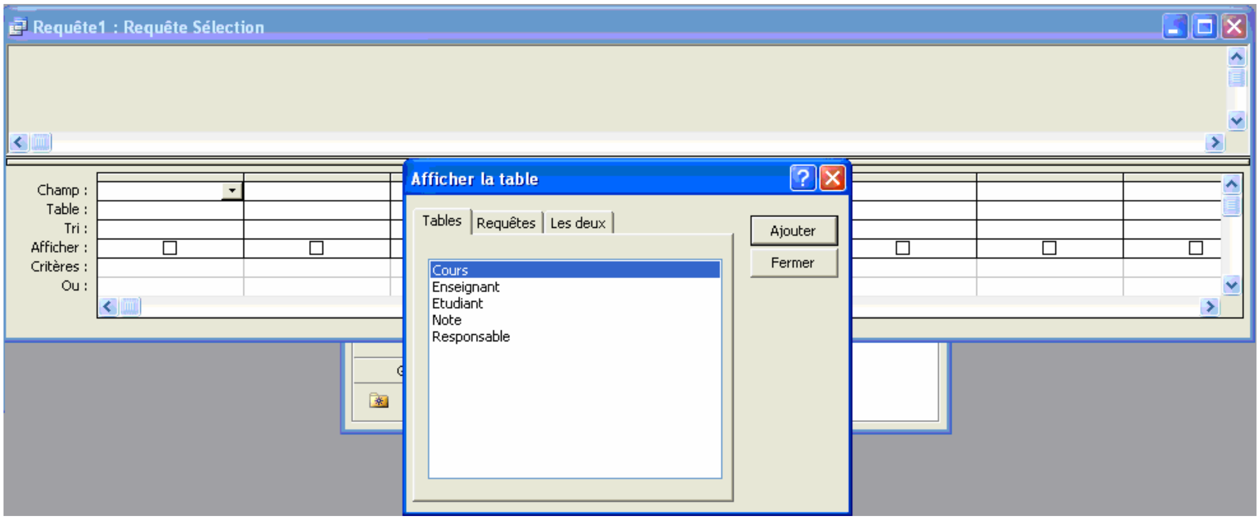


Figure 9: home page of the construction of queries

From the interface of the previous figure, one must double-click on each table required by the query. For example, to construct the first query “Extraction of all female students who have a mark greater than 12 in Information System in 2009”, the participating tables are “student”, “course” and “mark”. For each attribute used in the query, its corresponding table must first be selected in the line “table” before selecting itself in the line “field”. The option of the line “display” is used to specify if the values of the attribute must be displayed or not. The line “criterion” specifies the selection conditions (for example: mark>=12 and sex=”female”). The following figure presents the QBE query.

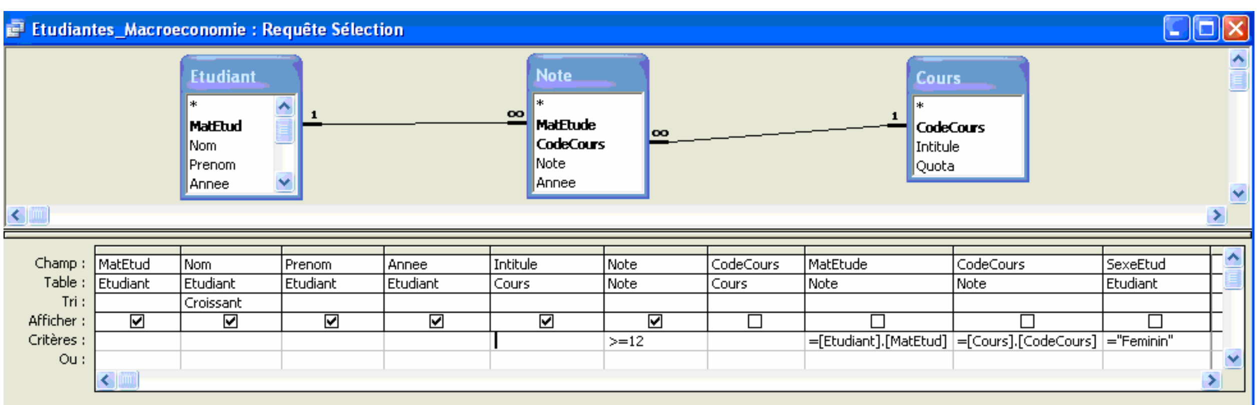


Figure 10: Example of QBE query

According to the previous figure, the result will be a table with 6 columns. Other columns are used to establish the selection criteria. Other queries are constructed with the same procedure.

4.2.4.2 The Structured Query Language (SQL)

SQL is fundamental and almost universal for all data base management system. It allows solving complex problems. The previous query can be translated in SQL as follow:

```
SELECT Student.StudentNum, Student.Name, Student.Surname,
Student.Year, Course.Name, Mark.Mean
FROM Course INNER JOIN (Student INNER JOIN [Mark] ON
Student.SudentNum =Mark.StudentNum) ON Course.CourseNum
=Mark.CourseNum
WHERE (((Mark.Mean)>=12) AND
((Mark.StudentNum)=[Student].[StudentNum]) AND
((Mark.CourseNum)=[Course].[CourseNum]) AND
((Student.StudentSex)="Female"))
ORDER BY Student.Name;
```

Figure11: SQL Code of the first query

For the second query of the expected results, the equivalent SQL code is the following:

```
SELECT Teacher.TeaCode, Teacher.TeaName, Teacher.TeaSurname
FROM Teacher INNER JOIN (Course INNER JOIN Principal ON
Course.CourseNum = Principal.CourseNum) ON Teacher.TeaCode =
Principal.TeaCode
WHERE (((Principal.CourseNum)=[Course].[CourseNum]) AND
((Principal.TeaCode)=[Teacher].[TeaCode]) AND
((Course.Name)="Data Base"));
```

Figure 12: SQL Code of the second query

To lunch the SQL editor, the procedure is similar to the one leading to the QBE editor. The unique difference lies in the fact that there is no table selection. The following page is then obtained.

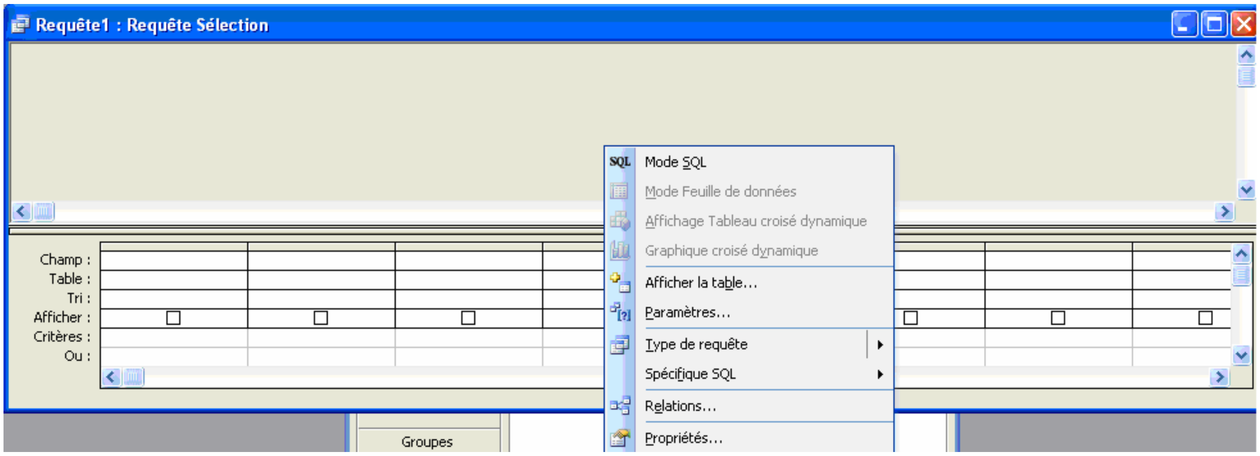


Figure 13 : *Sélection of the SQL mode*

From the previous figure, select the option "SQL Mode" to display the following page.

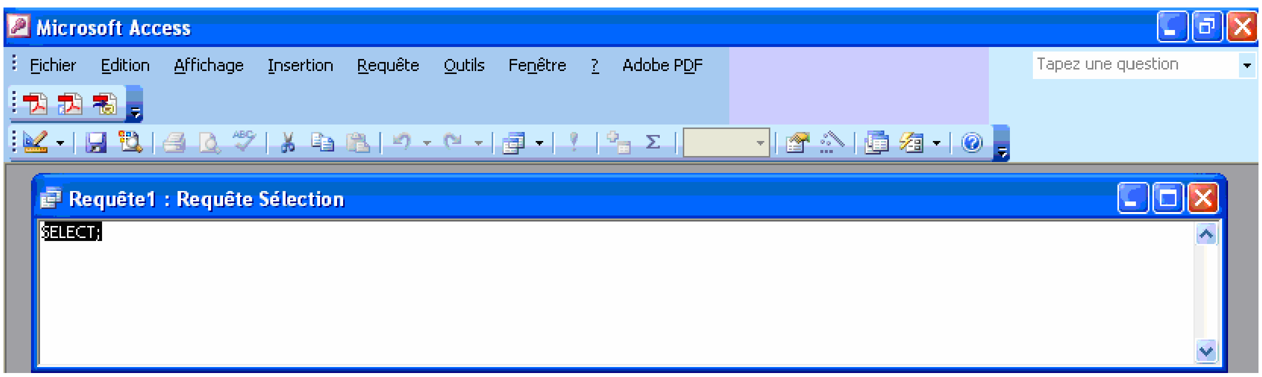


Figure 14: *Access SQL Editor*

In the SQL editor, one can edit SQL queries and execute or save under a specific name. One can also paste a SQL query copied elsewhere and execute.

4.2.5. Creation of forms

Forms are interfaces of interaction with the data base. Through forms, one can lunch queries, handle data of the data base and handle other forms.

To create a form, select the object "Form" in the column of objects and select the function "Create a form with wizard". Then the following page appears:

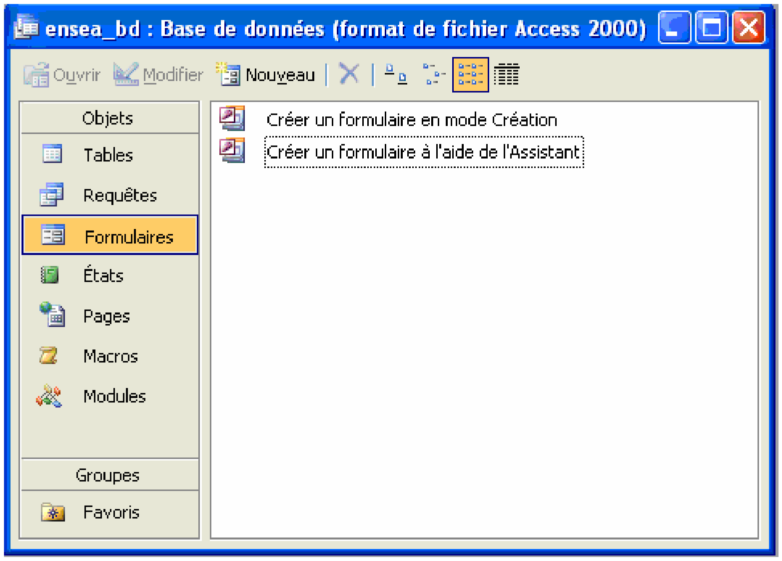


Figure 15: *Form creation page*

By double-clicking on the option “Create a form with wizard”, the following page appears:

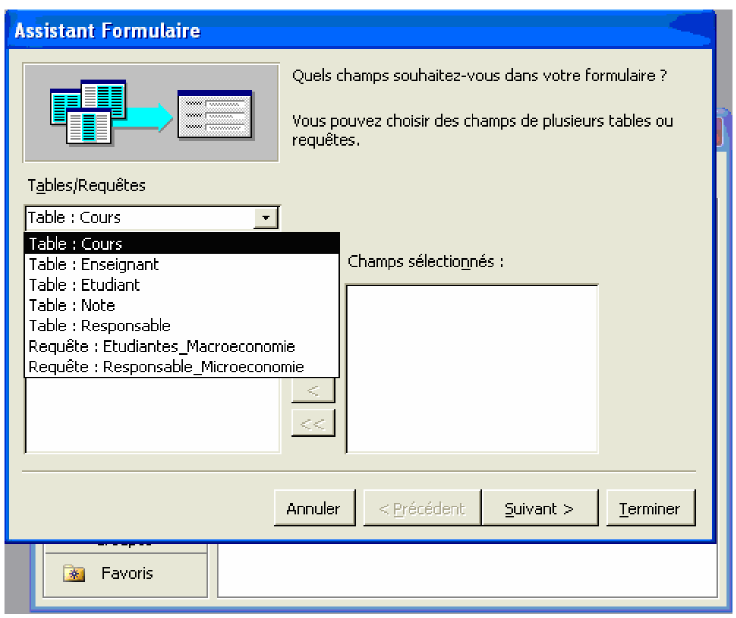


Figure 16: *Selection of attributes*

From the previous figure 16, select the tables or queries to integrate in the form. Sequentially, for each table or query selected, its attributes are displayed in the appropriate box. The selection of an attribute is done by double-clicking on it. For example, to fill the table “Student”, by selecting the table Student from the figure 16, the following page appears:

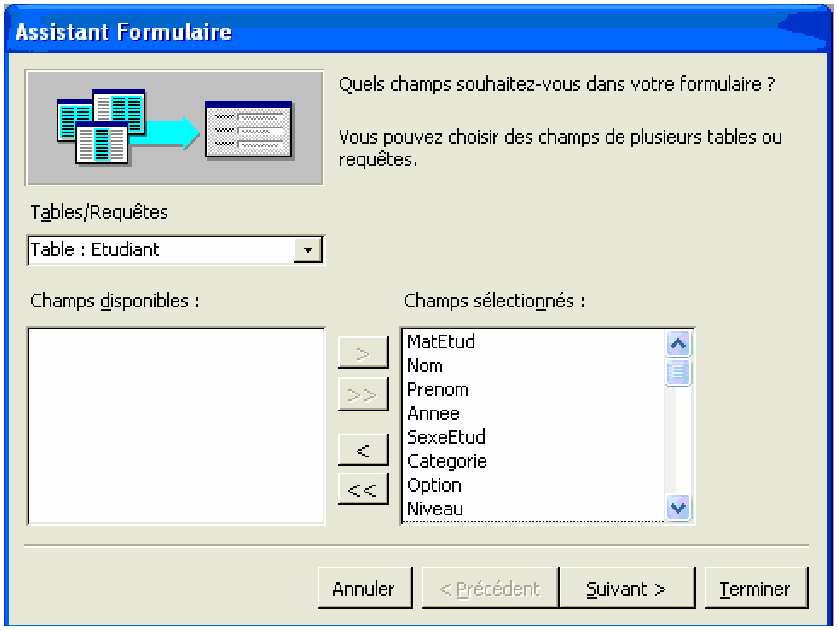


Figure 17: Selection of attributes of the table "Student" to fill from the form

By clicking on the button "Next", the following page appears:

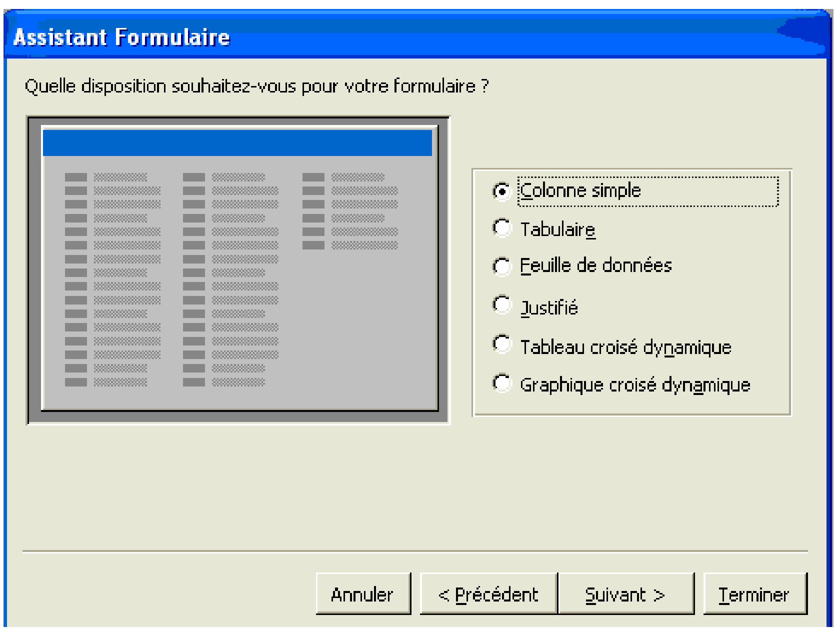


Figure 18: Form presentation

By choosing the option "Simple column" in figure 18, one can choose the style of form presentation in the following page.

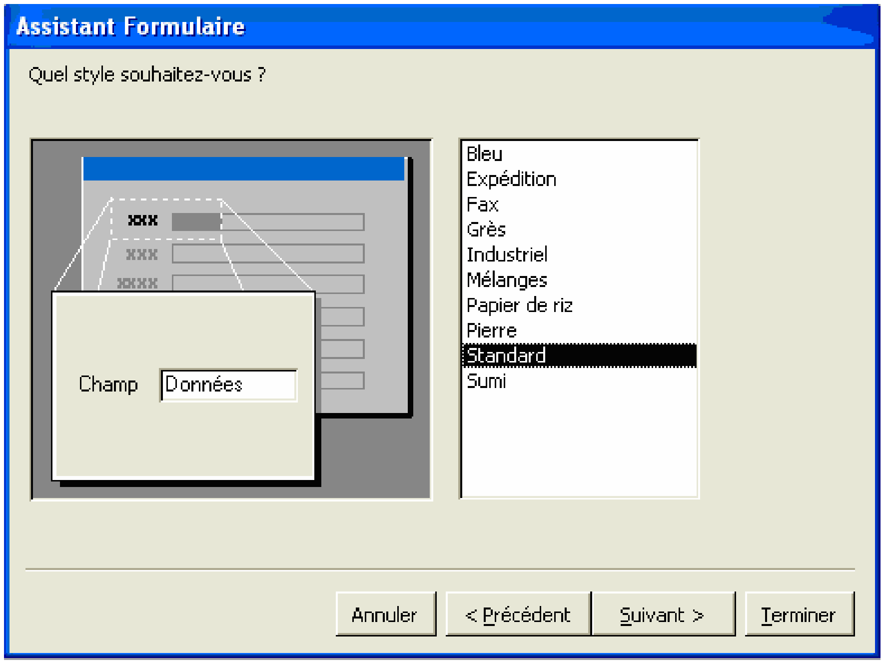


Figure 19: Choice of the form presentation.

By choosing the standard option, the following page is displayed:

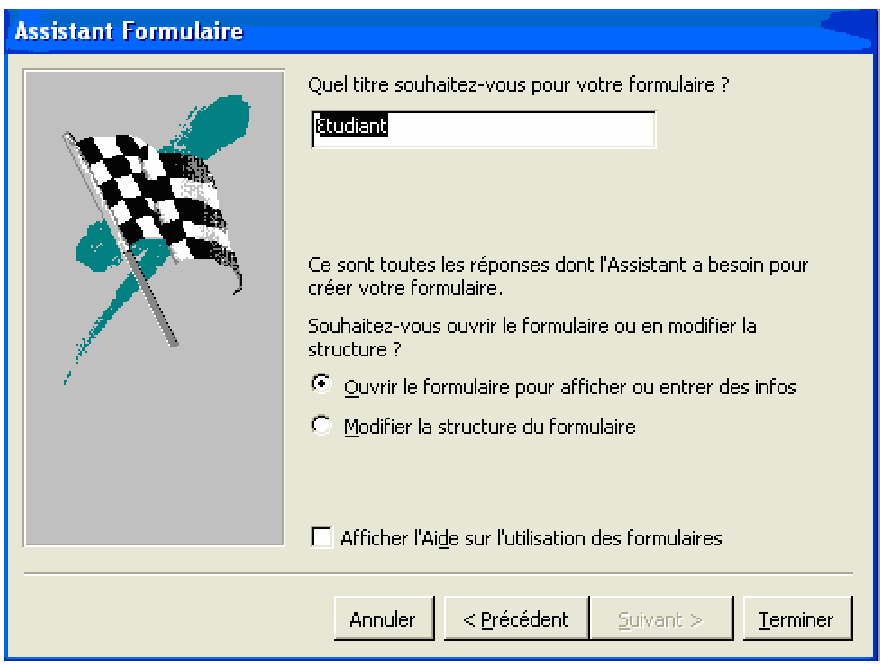


Figure 20: Saving Form Window.

Edit the name (Student) of the form in the appropriate field and click on the button "End" to display the final form "Student".

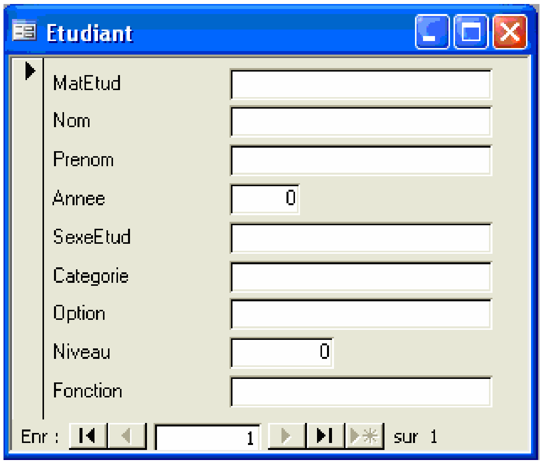


Figure 21: *Form "Student".*

By accessing this form in the creation mode, one can modify the position, the font style and the size of attributes and fields on the form.

4.2.6. Creation of macros

Macros are small functions (programs) integrated in Ms Access. Each macro has a specific task (opening or closing a form or a table, etc.) To create a macro, select the object "Macro" in the column of objects and click on the button "New" to display the following page:

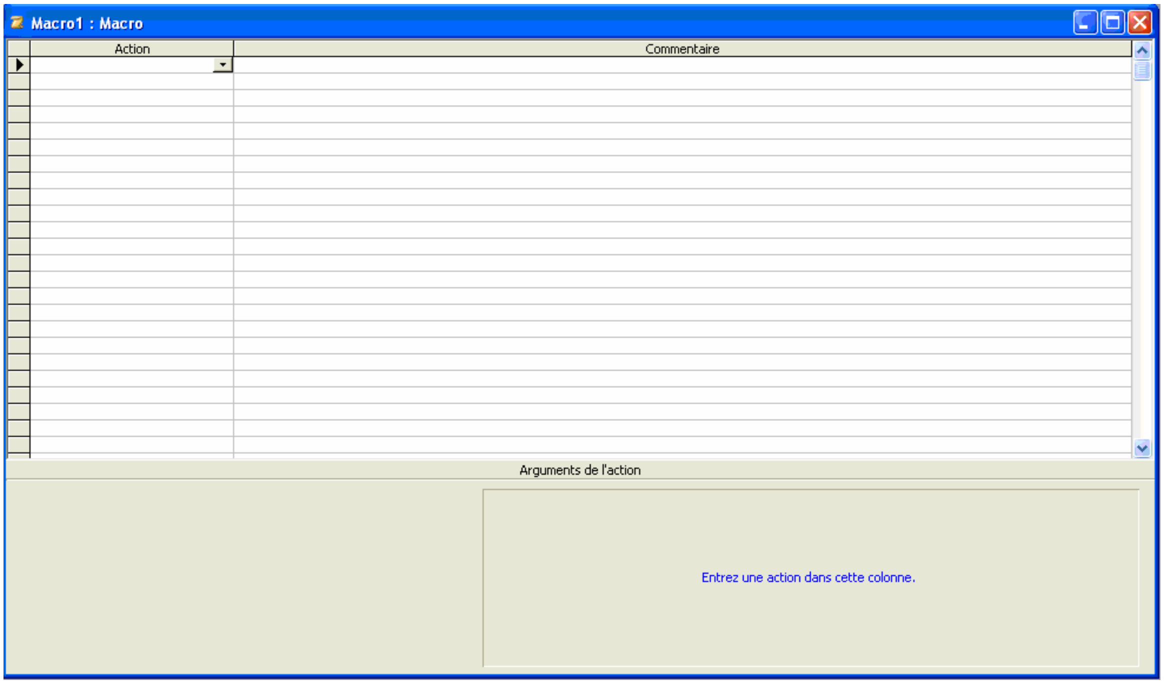


Figure 22: *Macro creation form.*

The column Action specifies different actions to integrate in the macro. The same macro can contain many actions. In this case actions are successively executed. For example, for a macro that will open the Form student The previous form is filled as follow:

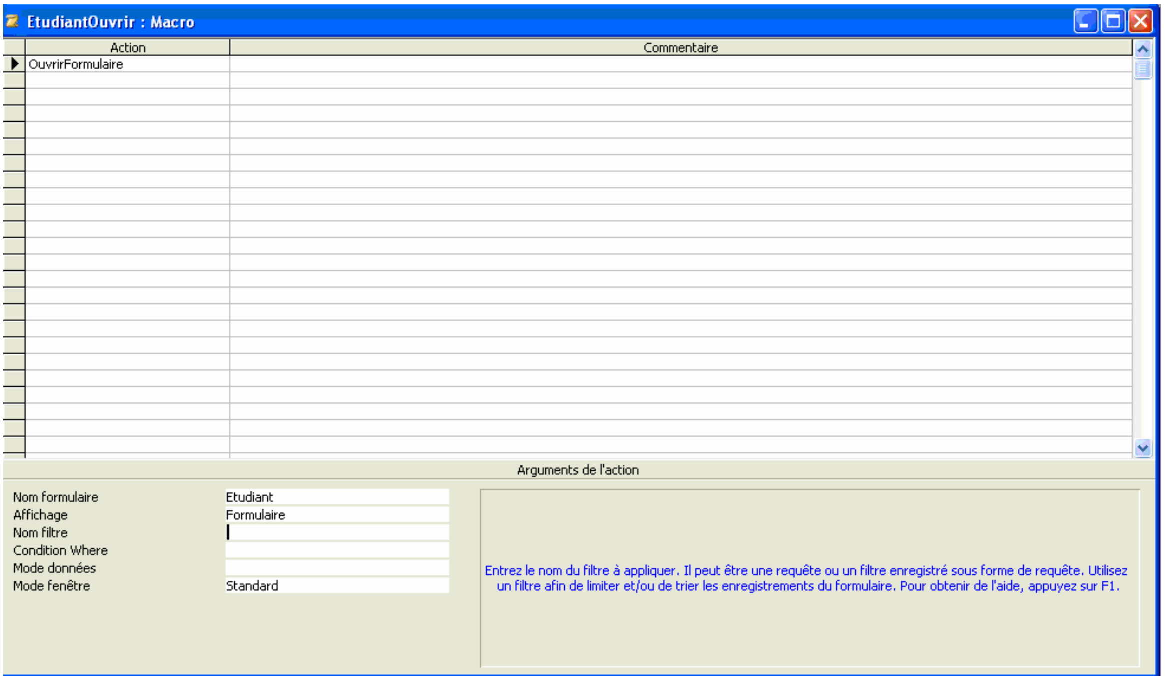


Figure 23: Example of macro opening the form "Student".

BIBLIOGRAPHIE

[1] Jean-Claude. COURBON : 1997, Dunod, "Systèmes d'information : structuration, modélisation et communication",288 pages, ISBN : 2729604227.

[2] André MULLER, 2005, Presses Universitaires de France (PUF), "L'informatique dans l'entreprise", 127 pages, 2ème édition, ASIN : 2130445926.

[3] Steven ALTER, 2001, Prentice Hall, "Information Systems: Foundation of E-Business", 587 pages 4ème édition, ISBN: 0130617733.

Quelques sites web utiles.

[a] http://www.isworld.org/