

LE MODELE LOGIQUE DE TYPE CODASYL

SANE ARNAUD & Dr N'GUESSAN GERARD

Table des matières



I - Objectifs	3
II - DEFINITION DES CONCEPTS DU MODELE LOGIQUE DE TYPE CODASYL	4
III - EVALUATION 1	6
IV - REGLES DE REALISATION DU MODELE CODASYL	7
V - EVALUATION 2	9
VI - Ressources annexes	10
VII - Solutions des exercices	12



Objectifs

- Définir les concepts du Modèle Logique de Type CODASYL
- Réaliser correctement le schéma du Modèle Logique de Type CODASYL

DEFINITION DES CONCEPTS DU MODELE LOGIQUE DE TYPE CODASYL



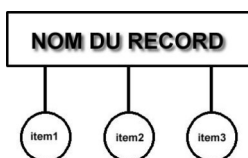
Définition

- c'est un modèle évolué du modèle hiérarchique.
- Il utilise des pointeurs entre les enregistrements mais ne tient pas compte de la hiérarchie des données.
- Ce modèle a été défini par le groupe de travail DBTG (Data Base Task Group) de la CONFERENCE ON DATA SYSTEME LANGUAGE.

1. CONCEPTS DU MODELE LOGIQUE DE TYPE CODASYL

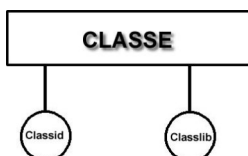
- *Record* : C'est un ensemble de données élémentaires représentant un enregistrement. Il existe deux types de record à savoir le record OWNER (propriétaire) et le record MEMBER (membre).
- *Data Item* : C'est une propriété du record
- *Set* : Il permet de mettre en relation deux records

2. FORMALISME D'UN RECORD



Vue plus grande - p.10

3. EXEMPLE DE RECORD

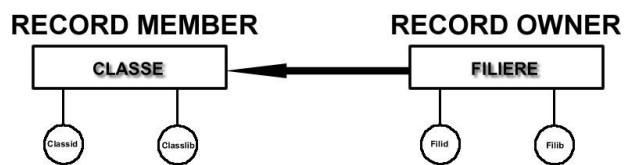


Vue plus grande - p.10

Dans cet exemple,

- le nom du record est *CLASSE*
- Classid (pour l'identifiant de la classe) et Classlib (pour le libellé de la classe) représentent les *items*

4. EXEMPLE DE RELATION ENTRE DEUX RECORDS



Vue plus grande - p.10

Dans ce exemple, nous avons une relation (ou un set) entre deux records à savoir : le record CLASSE et le record FILIERE

EVALUATION 1



Exercice 1

[Solution n°1 p 12]

Dans le Modèle Logique de données de Type Codasyl :

- Un ensemble de données élémentaires représentant un enregistrement est appelé .
- Une relation est appelée .
- Un élément d'un ensemble de données élémentaires est appelé .

Exercice 2

[Solution n°2 p 12]

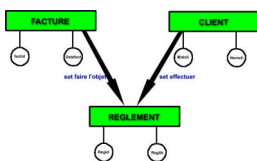
Cochez la bonne réponse

- ☐ Un set peut contenir plusieurs record
- ☐ Un record ne peut contenir qu'un seul item
- ☐ Un record peut disposer de plusieurs items

REGLES DE REALISATION DU MODELE CODASYL

1. PRINCIPES D'ELABORATION DU MODELE CODASYL

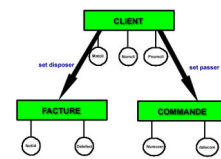
REGLE 1 : Tout record peut être membre de plusieurs sets différents.



Dans cet exemple, le record *REGLLEMENT* est à la fois membre des records *FACTURE* et *CLIENT*.

- Une facture peut faire l'objet de plusieurs règlements.
- Un client peut également effectuer plusieurs règlements

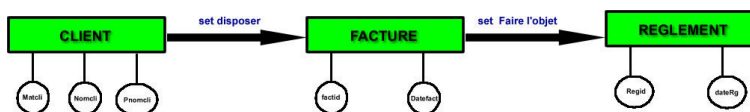
REGLE 2 : Tout record peut être propriétaire de plusieurs sets différents



Dans cet exemple, le record *CLIENT* est propriétaire des records *FACTURE* et *COMMANDE*.

- Un client peut disposer de plusieurs factures
- Un client peut également passer plusieurs commandes

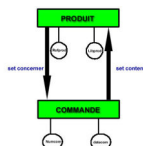
REGLE 3 : Tout record peut être à la fois propriétaire et membre de sets différents.



Dans cet exemple, le record *FACTURE* est à la fois membre du record *CLIENT* et propriétaire du record *REGLLEMENT*.

- Un client peut disposer de plusieurs factures.
- Une facture peut faire l'objet de plusieurs règlements

REGLE 4 : Deux records peuvent être reliés à un nombre quelconque de sets.



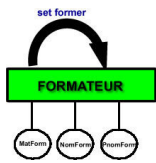
Dans cet exemple, les deux records sont mutuellement propriétaires et membres.

- Une commande peut contenir plusieurs produits, de ce fait, le record *COMMANDE* est

propriétaire du record *PRODUIT*

- De même, un produit peut se retrouver dans plusieurs commandes, de ce fait, le record *PRODUIT* devient propriétaire du record *COMMANDE*.

REGLE 5 : Un record peut être propriétaire et membre d'un même set.



Dans cet exemple, le record FORMATEUR est à la fois record membre et record propriétaire de lui même, du fait de la récursivité.

Un formateur peut former d'autres formateurs.

2. LES REGLES DE PASSAGE DU MCD AU MODELE CODASYL

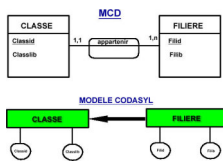
Le passage du Modèle Conceptuel des Données au Modèle Logique de Type Codasyl obéit à un certain nombre de règles à savoir :

- Chaque entité du Modèle Conceptuel des Données devient un Record
- L'identifiant devient une clé du Record
- Les propriétés deviennent des Data items.

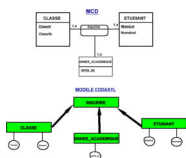
Exemple



- Une relation de type Père-fils devient un set partant du père c'est à dire du record OWNER (record propriétaire) au fils c'est à dire au record MEMBER (record membre).



- Une relation ternaire se transforme en record MEMBER en générant autant de sets qu'il y a d'entités.



EVALUATION 2

IV

Exercice 1

[Solution n°3 p 12]

Le passage du MCD au Modèle Logique de Type Codasyl transforme :

- ☐ L'entité père en record MEMBER et l'entité fils en record OWNER
- ☐ L'entité père en record OWNER et l'entité fils en record MEMBER
- ☐ L'entité père en record OWNER et en record MEMBER

Exercice 2

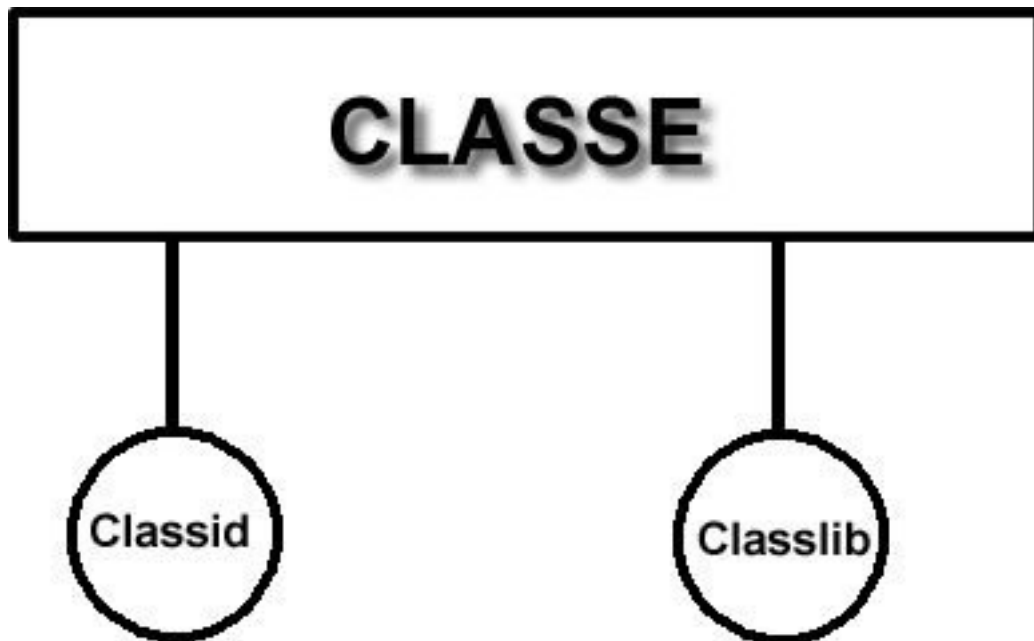
[Solution n°4 p 12]

La relation Père - Père (ou plusieurs à plusieurs) du MCD génère dans le CODASYL

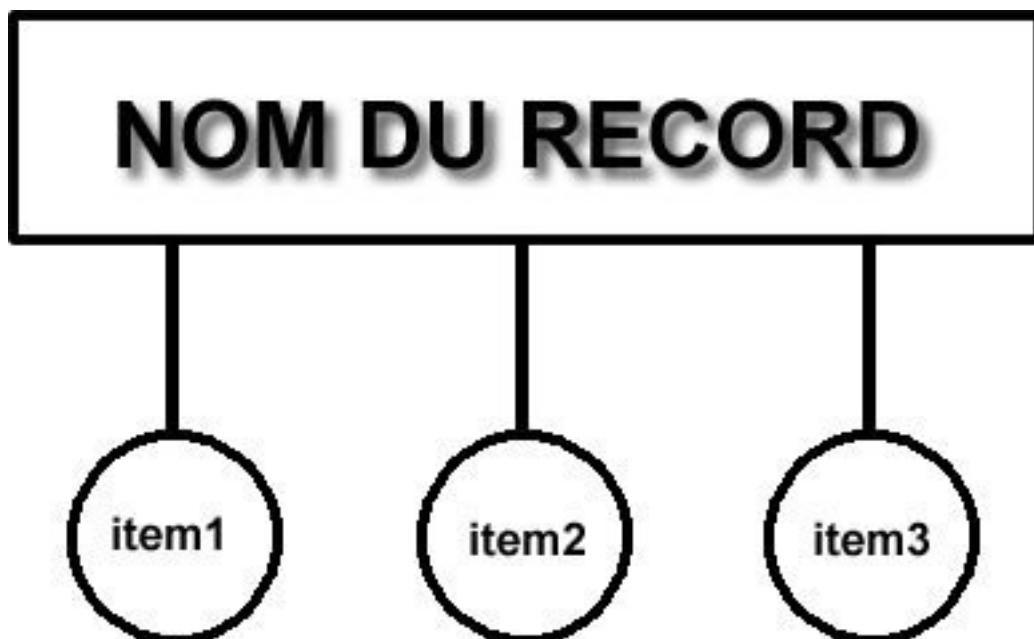
- ☐ Trois records dont les deux entités deviennent des records MEMBER et la relation devient un record OWNER
- ☐ Que deux records OWNER
- ☐ trois records dont les deux entités deviennent des records OWNER et la relation devient un record MEMBER

Ressources annexes

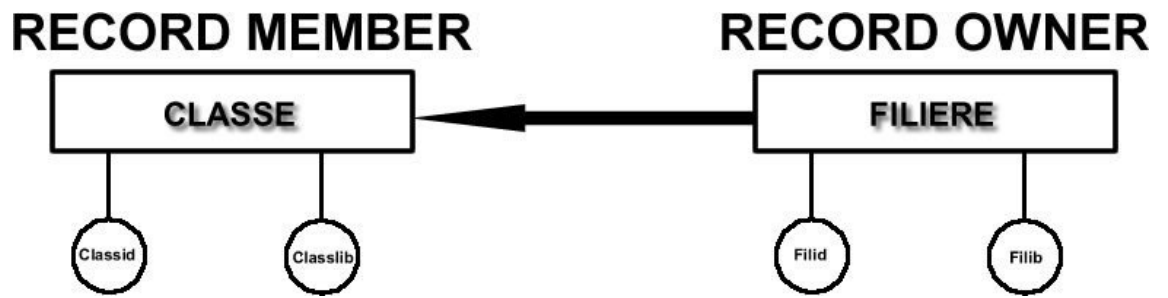
> EXEMPLE RECORD.jpg



> FORMALISME RECORD.jpg



> **RELATION RECORD.jpg**



Solutions des exercices



> Solution n° 1

Exercice p. 6

Dans le Modèle Logique de données de Type Codasyl :

- Un ensemble de données élémentaires représentant un enregistrement est appelé **record** .
- Une relation est appelée **set** .
- Un élément d'un ensemble de données élémentaires est appelé **item** .

> Solution n°2

Exercice p. 6

- ☐ Un set peut contenir plusieurs record
- ☐ Un record ne peut contenir qu'un seul item
- ☒ Un record peut disposer de plusieurs items

> Solution n°3

Exercice p. 9

- ☐ L'entité père en record MEMBER et l'entité fils en record OWNER
- ☒ L'entité père en record OWNER et l'entité fils en record MEMBER
- ☐ L'entité père en record OWNER et en record MEMBER

> Solution n°4

Exercice p. 9

- ☐ Trois records dont les deux entités deviennent des records MEMBER et la relation devient un record OWNER
- ☐ Que deux records OWNER
- ☒ trois records dont les deux entités deviennent des records OWNER et la relation devient un record MEMBER

