IFT3030
Base de données

Chapitre 3
Les modèles de données

- Introduction
- Architecture
- **Modèles de données**
- Modèle relationnel
- Algèbre relationnelle
- SQL
- Conception
- Fonctions avancées
- Concepts avancés
- Modèle des objets
- BD à objets

Plan du cours

- Introduction
- Architecture
- **Modèles de données**
- Modèle relationnel
- Algèbre relationnelle
- SQL
- Conception
- Fonctions avancées
- Concepts avancés
- Modèle des objets
- BD à objets

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Types de modèles

- Modèles conceptuels
  - Entités-Relations
  - UML
  - …
- Modèles de bases de données
  - Réseau
  - Hiérarchique
  - Relationnel
  - À objets
  - …

Chapitre 7

Ici

Chapitre 4

Chapitre 10

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, définitions

- **Atome** (Data Item) :

  plus petite unité de données possédant un nom

  Exemples : MATRICULE, NOM, …

  possède un type (« entier » pour MATRICULE)

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, définitions

- Objets
  - **Groupe** (Data aggregate) :

    collection d 'atomes rangés consécutivement dans la base et possédant un nom

  - Simple : suite d 'atomes

    Exemple ADRESSE formé de NUM, RUE, VILLE et CP)

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, définitions

- Objets
  - **Groupe** (Data agregate) :
  - Répétitif : collection de données apparaissant plusieurs fois consécutivement formé d 'atomes et/ou de groupe répétitifs
  - Exemple
    - PRENOM appelé **vecteur,**
    - ENFANT formé de PRENOM, SEX, AGE

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, définitions

- Objets
  - **Article** (Record) :

collection d 'atomes et de groupe rangés côte à côte dans la base de données

unité d 'échange entre la base et les applications

les occurrences d 'articles sont rangées dans des fichiers (AREA)

7

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

- Objets
  - Article

RECORD NAME IS <un nom>;
RECORD NAME IS EMPLOYES;

  - Atome

[<niveau>] <nom-atome> **TYPE IS** <spec-type>;
02 NOM **TYPE IS CHARACTER 12;**

  - Groupe

[<niveau>] <nom-groupe>
02 ADRESSE
03 RUE **TYPE IS CHARACTER 30;**
03 CODE **TYPE IS CHARACTER 7;**
03 VILLE **TYPE IS CHARACTER 20;**

8

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

- Objets
  - Type de données

décimal signé condensé ou non
SIGNED [{PACKED | UNPACKED}] DECIMAL <n1>, [<n2>]
entier binaire long (s+31bits) ou court (s+15bits)
SIGNED BINARY {15 | 31}
chaîne de caractère de longueur fixe (en caractères)
CHARACTER <n1>

  - Répétition des groupes

OCCURS <n1> **TIMES**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

- Exemple

RECORD NAME IS VINS**;**
02 NV **TYPE IS SIGNED PACKED DECIMAL 5;**
02 CRU **TYPE IS CHARACTER 10;**
02 MILLESIMES **OCCURS 5 TIMES**
03 ANNEE **TYPE IS SIGNED UNPACKED DECIMAL 4;**
03 DEGRE **TYPE IS BINARY 15 ;**

RECORD NAME IS PRODUCTEURS**;**
02 NP **TYPE IS SIGNED PACKED DECIMAL 5;**
02 NOM **TYPE IS CHARACTER 10;**
02 PRENOM **TYPE IS CHARACTER 10;**
02 ADRESSE
03 RUE **TYPE IS CHARACTER 30;**
03 CODE **TYPE IS CHARACTER 7;**
03 VILLE **TYPE IS CHARACTER 20;**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, définitions

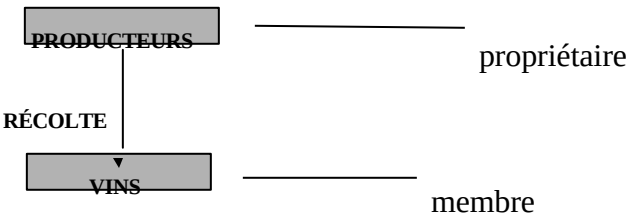
- Liens
  - **Lien** (set)

  type d 'association orientée entre articles de type T1 vers articles de type T2 dans laquelle une occurrence relie un article propriétaire de type T1 à n articles membres de type T2

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

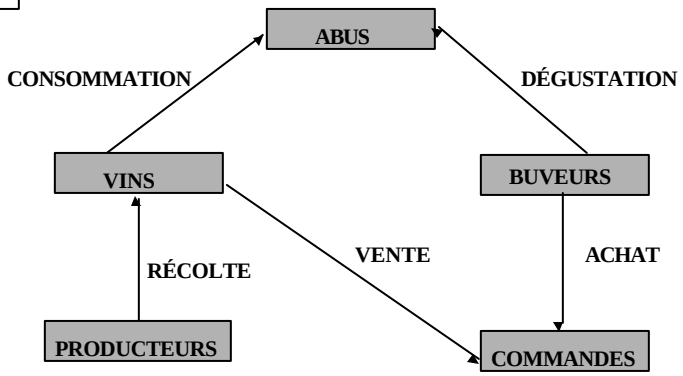
Modèle réseau, définitions

- Liens
  - **Diagramme de Bachman**



Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

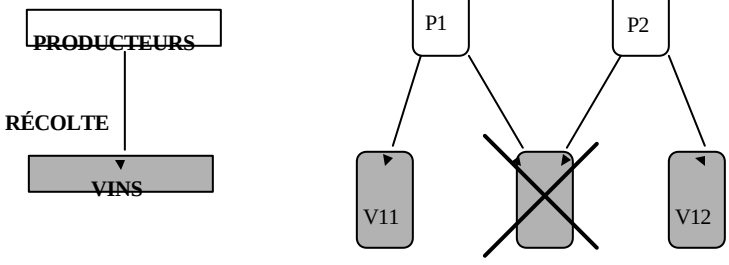
Modèle réseau, exemple



13

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, limites

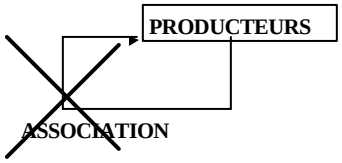


14

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

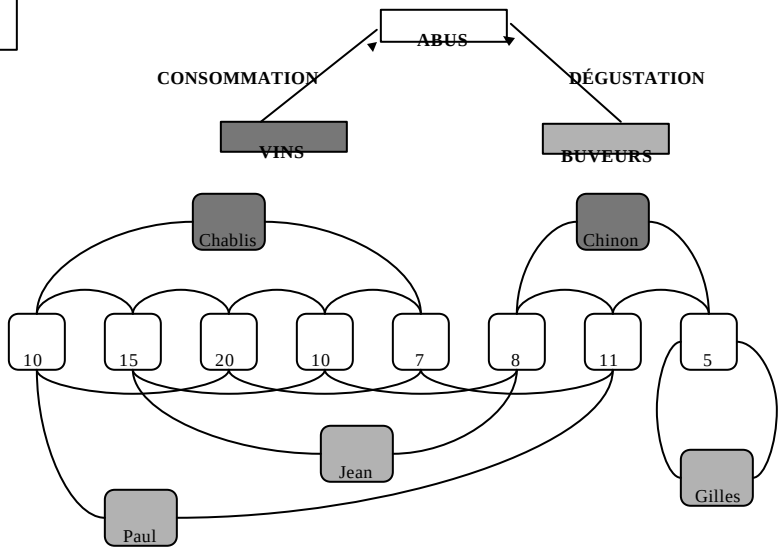
Modèle réseau, limites

- Pas de lien réflexif



Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, graphe des occurrences



Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

- liens

SET NAME IS <nom_de_lien>;
OWNER IS <nom_article>;
[**MEMBER IS** <nom_article>;]*

Exemple

SET NAME IS RECOLTE;
OWNER IS PRODUCTEURS;
MEMBER IS VINS;

Cas des liens singulier

Spécifier : **OWNER IS SYSTEM**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

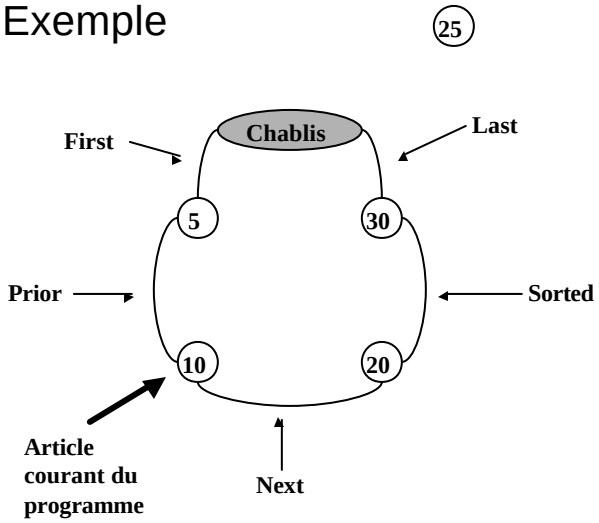
Modèle réseau, ordonnancement

- Les articles dans les occurrences de liens sont ordonnés pour l 'insertion
- Positions
  - début (FIRST) ou fin (LAST)
  - avant (PRIOR) ou après (NEXT)
  - par tri (SORTED) croissant ou décroissant d 'un atome (KEY)

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, ordonnancement

■ Exemple



19

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

■ Ordonnancement

ORDER IS [PERMANENT] INSERTION IS
{FIRST | LAST | PRIOR | NEXT | SORTED <spécification de tri>;

spécification de tri
[**RECORD-TYPE SEQUENCE IS** <nom_article> $^{+}$]
BY DEFINED KEYS [DUPLICATES ARE {FISRT | LAST | NOT ALLOWED}]

définition de clé dans article membre
KEY IS {ASCENDING | DESCENDING} <nom de donnée>

20

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

- Exemple

SET NAME IS RECOLTE;
OWNER IS PRODUCTEURS;
ORDER IS PERMANENT INSERTION IS SORTED
BY DEFINED KEYS DUPLICATES ARE FISRT ;
MEMBER IS VINS;
KEY IS ASCENDING CRU ;

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, sélection

- Il y a autant d ’occurrences de type de lien que d ’articles propriétaires
- sélection de l ’occurrence de type de lien
  - manuelle dans le programme
  - automatique par la clause SET SELECTION

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

- sélection

SET SELECTION [FOR <nom_lien1>] **IS**
THRU <nom_lien2> **OWNER IDENTIFIED BY**
{ APPLICATION
| DATA-BASE-KEY [EQUAL TO <nom_paramètre1> **]**
| CALC KEY [EQUAL TO <nom_paramètre2> **]}**
[THEN THRU <nom_lien3> **WHERE OWNER IDENTIFIED BY**
{<nom_Donnée3> **[EQUAL TO** <nom_paramètre3> **]}** $^{+}$ **] ...**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, options d'insertion

- Lors de la création d 'un nouvel article dans la base, il pourra être inséré dans un lien de manière
  - automatique
  - manuelle
- Contrainte
  - obligatoire
  - facultative
- Syntaxe

INSERTION IS {AUTOMATIC | MANUAL}
RETENTION IS {MANDATORY | OPTIONAL}

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, placement des articles

- BD réseau est stockée dans un ensemble de fichiers appelés AREA ou REALM
  - fichiers relatifs

    adresse = numéro de page et numéro d 'octet dans la page
  - fichiers aléatoires

    adresse = *f(c)*, fonction de hachage
- Clé de base de données (database key)

  Adresse invariante affectée à un article lors de sa création et permettant de l 'identifié sans ambiguïté

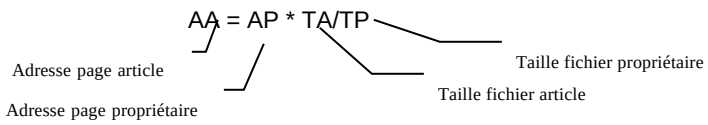
  exemple : num_fichier num_page deplacement

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, placement des articles

- Placement CODASYL

  Méthode de calcul de l 'adresse d 'un article et d 'attribution de la clé de base de données lors de la première insertion
  - placement direct (DIRECT)
  - placement calculé (CALC USING)
  - placement par lien (VIA)
    - par proximité (même fichier que propriétaire)
    - par homothétie (autre fichier)



Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, syntaxe

■ Placement des articles

LOCATION MODE IS
{ SYSTEM
| DIRECT <nom_paramètre>
| **CALC USING** <nom_donnée>
[DUPLICATES ARE [NOT] ALLOWED]
| VIA <nom_lien> **SET}**
WITHIN {<nom_fichier> **| AREA OF OWNER}**

27

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, exemple

SCHEMA NAME IS VINICOLE
AREA NAME IS F-BUVEURS ;
AREA NAME IS F-PRODUCTEURS ;
AREA NAME IS F-COMMADES ;

RECORD NAME IS BUVEURS **;**
LOCATION MODE IS CALC USING NB **DUPLICATES**
NOT ALLOWED WITHIN F-BUVEURS ;
02 NB **TYPE IS SIGNED PACKED DECIMAL 5;**
02 NOM **TYPE IS CHARACTER 10;**
02 PRENOM **TYPE IS CHARACTER 10;**

RECORD NAME IS ABUS**;**
LOCATION MODE IS VIA DEGUSTATION
WITHIN AREA OF OWNER ;
02 QUANTITE **TYPE IS SIGNED BINARY 15;**

28

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, exemple

```
RECORD NAME IS PRODUCTEURS ;
    LOCATION MODE IS CALC USING NOM DUPLICATES
    ALLOWED WITHIN F-PRODUCTEURS;
    02 NOM TYPE IS CHARACTER 10 ;
    02 REGION TYPE IS CHARACTER 8 ;

RECORD NAME IS VINS ;
    LOCATION MODE IS VIA RECOLTE
    WITHIN AREA OF OWNER ;
    02 NV TYPE IS SIGNED PACKED DECIMAL 5 ;
    02 CRU TYPE IS CHARACTER 10;
    02 MILLESIMES OCCURS 5 TIMES ;
        03 ANNEE TYPE IS SIGNED UNPACKED DECIMAL 4;
        03 DEGRE TYPE IS BINARY 15 ;
```


Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, exemple

Compatible an 2000

```
RECORD NAME IS COMMANDES;
    LOCATION MODE IS VIA ACHAT WITHIN F-COMMANDES ;
    02 DATE TYPE IS CHARACTER 8 ;
    02 QUANTITE TYPE IS SIGNED BINARY 15 ;

SET NAME IS DEGUSTATION;
    OWNER IS BUVEURS;
    ORDER IS PERMANENT INSERTION IS LAST ;
    MEMBER IS ABUS;
    INSERTION IS AUTOMATIC RETENTION IS MANDATORY ;
    SET SELECTION FOR DEGUSTATION IS
    THRU DEGUSTATION OWNER IDENTIFIED BY CALC KEY ;
```

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, exemple

SET NAME IS CONSOMMATION;
OWNER IS VINS;
ORDER IS PERMANENT INSERTION IS NEXT ;
MEMBER IS ABUS;
INSERTION IS AUTOMATIC RETENTION IS MANDATORY ;
SET SELECTION FOR CONSOMMATION **IS**
THRU CONSOMMATION **OWNER IDENTIFIED BY APPLICATIONS ;**

SET NAME IS RECOLTE;
OWNER IS PRODUCTEURS;
ORDER IS PERMANENT INSERTION IS SORTED
BY DEFINED KEYS DUPLICATES ARE FISRT ;
MEMBER IS VINS;
INSERTION IS AUTOMATIC RETENTION IS OPTIONAL ;
KEY IS ASCENDING CRU ;
SET SELECTION IS THRU RECOLTE **OWNER IDENTIFIED BY CALC KEY ;**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, exemple

SET NAME IS VENTE;
OWNER IS VINS;
ORDER IS PERMANENT INSERTION IS SORTED
BY DEFINED KEYS DUPLICATES ARE NOT ALLOWED ;
MEMBER IS COMMANDES;
INSERTION IS AUTOMATIC RETENTION IS MANDATORY ;
KEY IS DESCENDING DATE **DUPLICATES NOT ALLOWED** ;
SET SELECTION IS THRU RECOLTE **OWNER IDENTIFIED BY CALC KEY**
THEN THRU VENTE **WHERE OWNER IDENTIFIED BY** NV **;**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, exemple

SET NAME IS ACHAT;
OWNER IS BUVEURS;
ORDER IS PERMANENT INSERTION IS LAST ;
MEMBER IS COMMANDES;
INSERTION IS AUTOMATIC RETENTION IS MANDATORY ;
SET SELECTION IS THRU ACHAT **OWNER IDENTIFIED BY APPLICATION ;**

END-SCHEMA.

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Sous-schéma COBOL
  - Sous-ensemble du schéma vu par un programme d 'application, spécifiant la vision externe de la base par un programme
  - On peut omettre
    - des types d 'articles, des fichiers, des liens et des données
  - On peut redéfinir
    - l 'ordre des atomes
    - le type d 'un atome
    - les clause SET SELECTION
    - les noms des types d 'articles, des atomes et des liens

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Sous-schéma COBOL
  - Se compose de
    - Division de titre
      - Nom
      - Schéma
    - Division de correspondance
      - synonymes
    - Division de structure
      - Fichiers
      - Articles
      - Liens

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Sous-schéma COBOL
  - Exemple

```
TITLE DIVISION
SS CLIENT WITHIN SCHEMA VINICOLE
MAPPING DIVISION
ALIAS SECTION
AD SET BOIT IS DEGUSTATION
AD SET ACHETE IS ACHAT
STRUCTURE DIVISION
REALM SECTION
RD F-BUVEURS
RD F-COMMADES
```

```
RECORD SECTION
01 BUVEURS
   02 NV PICTURE IS 999
   02 NOM PICTURE IS X(10)
   02 PRENOM PICTURE IS X(10)
01 ABUS
   02 QUANTITE PICTURE IS 999
01 COMMANDES
   02 QUANTITE PICTURE IS 999
   02 DATE PICTURE IS X(8)
SET SECTION
SD BOIT
SD ACHETE
```

Compatible an 2000

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Navigation CODASYL
  - Verbes de manipulation
    - recherche d 'articles (FIND)
    - échanges d 'articles (GET, STORE)
    - mise à jour (ERASE, CONNECT, DISCONNECT, MODIFY)
    - contrôle de fichiers (READY, FINISH)
  - Échanges via USER WORKING AREA
    - GET et STORE
    - Chaque atome ou article a une place fixe dans ce tampon

37

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

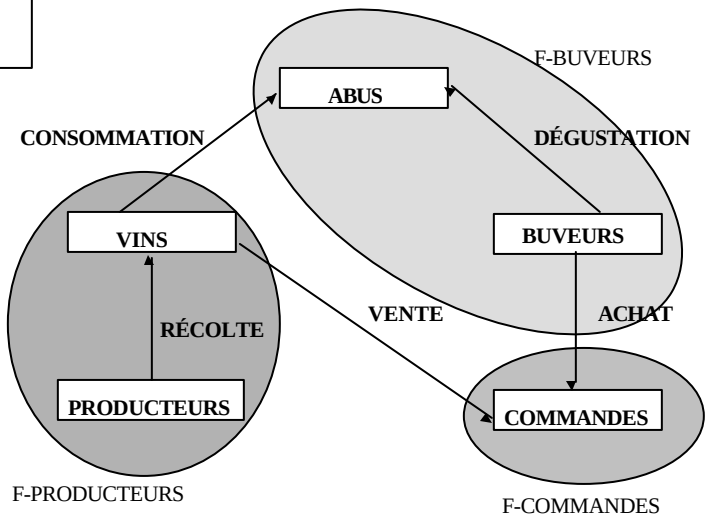
Langage de manipulation COBOL-CODASYL

- Navigation CODASYL
  - Curseur
    - Pointeur courant contenant la clé base de données du dernier article manipulé d 'une collection d 'articles, et permettant au programme de se déplacer dans la base
    - Plusieurs curseurs dans un programme
      - dernier article lu, écrit ou recherché
      - article courant de chaque type d 'article
      - article courant de chaque type de lien
      - article courant de chaque fichier
    - Seul un article pointé par un curseur d 'un pgm peut être lu par ce pgm
    - Nombre de curseurs = ??

38

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle réseau, exemple



Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Contrôle de fichiers
  - Ouverture

    READY {<nom fichier> [**USAGE-MODE IS**
    {EXCLUSIVE | PROTECTED} {RETRIEVAL | UPDATE}]}+**
  - Fermeture

    FINISH {<nom fichier>}+

- Échanger des articles
  - base vers la zone de travail

    GET {[<nom article>] | {<nom donnée>}*}
  - vers la base

    STORE <nom article> **RETAINING CURRENCY FOR**
    { MULTIPLE | REALM | RECORD | SETS | <nom de lien>+**}**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche d 'articles
  - l 'exécution du FIND permet de déplacer le curseur du programme mais pas forcément les autres
  - Syntaxe

FIND <expression de selection> **RETAINING CURRENCY FOR**
{ MULTIPLE
| REALM
| RECORD
| SETS
| <nom de lien>+**}**

41

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche sur clé de base de données
  - connaissant la clé BD

FIND <nom article> **DBKEY IS** <nom paramètre>

  - Connaissant la clé de hachage (unique)

FIND ANY <nom article>

    - clé doit être chargé par le pgm dans la zone de travail

  - Connaissant la clé de hachage (multiple)

FIND DUPLICATE <nom article>

    - plusieurs FIND DUPLICATE permettront de retrouver les doubles successivement

42

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche sur clé de base de données
  - Exemple
    - Accès sur clé de hachage (unique) au buveur dont le NB est 100

```
READY F-BUVEURS  USAGE-MODE IS PROTECTED RETRIEVAL.
MOVE `100´ TO NB  IN BUVEURS.
FIND ANY BUVEURS.
GET BUVEURS.
PRINT NOM, PRENOM IN BUVEURS.
FINISH F-BUVEURS.
```


Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche dans un fichier
  - séquentielle
    - 1er, dernier, suivant et précédent
  - directe
    - i^e article

FIND {FIRST | LAST | NEXT | PRIOR | <i> | <paramètre>**}**
<nom article> **WITHIN** <nom fichier>

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche dans un fichier
  - Exemple : imprimer tous les buveurs

```
READY F-BUVEURS.
FIND FIRST BUVEURS  WITHIN F-BUVEURS.
PERFORM UNTIL ``FIN-DE-FICHIER´´.
    GET BUVEURS.
    PRINT NOM, PRENOM IN BUVEURS.
    FIND NEXT BUVEURS  WITHIN F-BUVEURS.
END-PERFORM.
FINISH F-BUVEURS.
```


Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche dans une occurrence de lien
  - Idem

    FIND {FIRST | LAST | NEXT | PRIOR | <i> | <paramètre>**}**
    <nom article> **WITHIN** <nom lien>
  - à partir d ’une valeur de donnée

    FIND <nom article> **WITHIN** <nom lien>
    USING <nom donnée>+ — Unique/1e

    FIND DUPLICATE WITHIN <nom lien>
    USING <nom donnée>+ — Suivante
  - à partir d ’un article membre

    FIND OWNER WITHIN <nom lien>

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche dans une occurrence de lien
  - Exemple 1 : imprimer tous les buveurs et leurs commandes

```
READY F-BUVEURS, F-COMMANDES.
FIND FIRST BUVEURS  WITHIN F-BUVEURS.
PERFORM UNTIL ``FIN-DE-FICHIER´´
        GET BUVEURS.
        PRINT NOM, PRENOM IN BUVEURS.
        FIND FIRST COMMANDE  WITHIN ACHETE.
        PERFORM UNTIL ``FIN-D 'OCCURRENCE-DE-LIEN´´
                GET COMMANDES.
                PRINT QUANTITE, DATE IN COMMANDE.
                FIND NEXT COMMANDE  WITHIN ACHETE.
        END-PERFORM.
        FIND NEXT BUVEURS  WITHIN F-BUVEURS.
END-PERFORM.
FINISH F-BUVEURS, F-COMMANDES.
```

47

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Recherche dans une occurrence de lien
  - Exemple 2 : Imprimer les cmds du buveur 100 du 10/02/81

```
READY F-BUVEURS, F-COMMANDES.
MOVE `100´ TO NB  IN BUVEURS.
FIND ANY BUVEURS.
MOVE `10-02-81´ TO DATE  IN COMMANDES.
FIND COMMANDES WITHIN ACHETE USING DATE.
PERFORM UNTIL ``PLUS-DE-COMMANDES´´.
        GET COMMANDES.
        PRINT QUANTITE, DATE IN COMMANDE.
        FIND DUPLICATE WITHIN ACHETE USING DATE.
END-PERFORM.
FINISH F-BUVEURS, F-COMMANDES.
```

48

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Conditionnelles
  - tester si article courant appartient à un lien

IF [NOT] <nom lien> {**OWNER | MEMBER | TENANT}**
EXECUTE <instructions>

  - tester si une occurrence de lien n 'est pas vide

IF <nom lien> **IS [NOT] EMPTY EXECUTE** <instructions>

49

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Conditionnelles
  - Exemple :

```
READY F-BUVEURS, F-COMMANDES.
FIND FIRST ABUS WITHIN F-BUVEURS.
PERFORM UNTIL ``FIN-DE-FICHIER´´.
    GET ABUS.
    IF QUANTITE IN ABUS > 100 EXECUTE
        FIND OWNER WITHIN BOIT.
        IF ACHETE IS NOT EMPTY EXECUTE
            GET BUVEURS.
            PRINT NOM, PRENOM IN BUVEURS.
        END-IF.
    END-IF.
    FIND NEXT ABUS WITHIN F-BUVEURS.
FINISH F-BUVEURS.
```

50

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Positionner le curseur de programme
  - à celui d 'un type d 'articles, de lien ou de fichier

FIND CURRENT [<nom article>]
[WITHIN {<nom fichier> | <nom lien> }**]**

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Mises à jour
  - Suppression d 'articles

ERASE [ALL] [<nom article>]
- All : suppression récursive des membres
- Exemple : supprimer le buveur numéro 100

```
READY F-BUVEURS, F-COMMANDES.
MOVE `100´ TO NB IN BUVEURS.
FIND ANY BUVEURS.
ERASE ALL BUVEURS.
FINISH F-BUVEURS, F-COMMANDES.
```

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Mises à jour
  - Modification d 'articles

MODIFY {[<nom article>] | {<nom donnée>}*}
[INCLUDING {ALL | ONLY <nom lien>+} **MEMBERSHIP**]

- Exemple : ajouter 10 à la 7e quantité bue par le buveur 200

```
READY F-BUVEURS.
MOVE `7´ TO I.
MOVE `200´ TO NB  IN BUVEURS.
FIND ANY BUVEURS.
FIND I ABUS WITHIN BOIT.
GET ABUS.
ADD 10 TO QUANTITE IN ABUS.
MODIFY ABUS.
FINISH F-BUVEURS.
```


Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Langage de manipulation COBOL-CODASYL

- Mises à jour
  - insertion/suppression dans une occurrence de lien
    - Par le programme en mode MANUAL ou OPTIONAL
    - insertion article courant du programme

CONNECT [<nom article>] **TO** <nom lien>

- suppression article courant du programme

DISCONNECT [<nom article>] **FROM** <nom lien>}

- <nom article> permet de vérifier le type de l 'article courant

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle hiérarchique, généralités

- Vient de l 'idée que le monde réel apparaît souvent à travers des hiérarchies
- Cas particulier du modèle réseau
- Les liens forment des graphes hiérarchiques
- Absence de données répétitives

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle hiérarchique, concepts

- Champ (*Field*)
  - Plus petite unité de données possédant un nom (équivalent à un atome)
- Segment (*Segment*)
  - Collection de champs rangés consécutivement dans la base, portant un nom et dont une occurrence constitue l 'unité d 'échange entre la base et les applications (équivalent à un article sans groupe répétitif)

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle hiérarchique, concepts

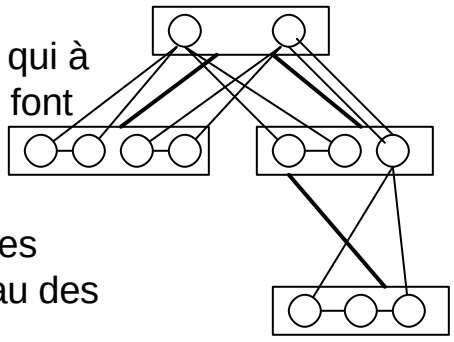
- Segment (suite)
  - De taille fixe
  - Tous les champs sont au même niveau
  - Peut posséder une clé (champ discriminant)
- Arbre de segments (*Segment tree)*
  - Collection de segment reliés par des associations père-fils, organisée sous la forme d 'une hiérarchie

57

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle hiérarchique, concepts

- Arbre de segments (suite)
  - liens de 1 vers N qui à un segment père font correspondre N segments fils
  - Tant au niveau des types qu'au niveau des occurrences



58

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

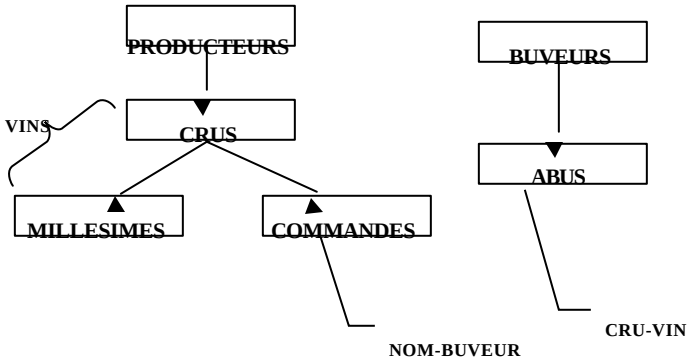
Modèle hiérarchique, concepts

- Base de données hiérarchique (Hierarchical database)
  - Base de donnée constituée par une forêt de segments
  - Un ensemble d 'arbres
  - Chaque arbre possède
    - un segment racine
    - des segments internes
    - des segments feuilles

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle hiérarchique, concepts

- Les liens ne sont pas nommés
- Exemple



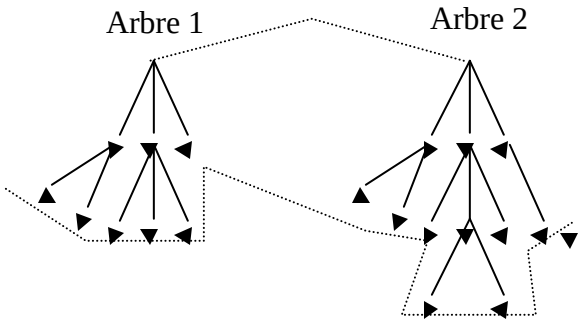
Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle hiérarchique, navigation

- Langages de manipulation des BD hiérarchiques
- Exemple DL1 pour IMS (IBM)
- Imbriqués dans langages de haut niveau (PL1, COBOL et FORTRAN)
- Ordre de navigation en profondeur d 'abord et de gauche à droite

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Modèle hiérarchique, navigation



- Curseurs (PCB) permettent de mémoriser un position dans la base

Introduction
Architecture
Modèles de données
Modèle relationnel
Algèbre relationnelle
SQL
Conception
Fonctions avancées
Concepts avancés
Modèle des objets
BD à objets

Référence

- Georges Gardarin, « Bases de données objet & relationnel », eds Eyrolles, Chapitre 4, 1999