INTRODUCTION AUX BASES DE DONNEES

16H Cours / 18H TD / 20H TP

1. INTRODUCTION

INFORMATIQUE : *ensemble des disciplines scientifiques et techniques spécialement applicables au traitement de l'information effectué notamment par des moyens automatiques* [définition AFNOR]

Codage de l'information

Toute information est représentée par un symbole.

Deux alphabets (ou codage) sont utilisés pour codifier les caractères (symboles d'information) sur la base de cet octet :

- le codage **ASCII** : le codage se fait sur 7 éléments binaires, soit $2^7 = 128$ codes distincts (le codage ASCII étendu se fait sur 8 bits et ce 8ème bit tient compte des caractères accentués) étant utilisé comme bit de contrôle.
- le codage **EBCDIC** : codage sur 8 éléments binaires. Soit $2^8 = 256$ codes distincts

Structures de données

Les ***structures de données*** requises pour décrire les diverses informations, c'est-à-dire les objets réels à traiter vont s'appuyer sur ces alphabets (ou codages).

Deux principaux modèles de structures de données :

- les systèmes de fichiers (lecture séquentielle de fichier : voir 1ier semestre)
- les systèmes de bases de données

Avant les années 80, approche descendante, chaque application crée ses données propres. D'où multiplication des **fichiers** et redondance des données.

A partir des années 80, approche ascendante, la **base de données** est conçue pour être utilisée par plusieurs applications.

Dans les deux cas on cherche à décrire des *entités* manipulés par une ou plusieurs applications.

Exemples d'entités: un client dans la gestion commerciale d'une entreprise
un étudiant dans la gestion de la scolarité de la faculté
une commande dans la gestion commerciale d'une société
un patient dans la gestion d'un cabinet médicale
.... etc

Une entité est caractérisée par un certain nombre de *propriétés* qui la définissent par rapport au type de traitements envisagés

Exemples :

pour le Client : nom, prénom, numéro de compte, débit, crédit, solde
pour l'étudiant : nom, prénom, numéro d'inscription, année de naissance, année d'étude, filière,...
...etc

L'ensemble des propriétés considérés définit en fait un *type d'entité*. Une entité particulière correspond à un ensemble de valeurs spécifiques pour ces propriétés.

Exemples : type d'entité : Client

Occurrence d'entité :

nom	prénom	N° de compte	débit	crédit	solde

DUPONT	Pierre	8610089	20.00	100.00	80.00

2. SYSTEMES DE GESTION DE BASES DE DONNEES

2.1. INTRODUCTION AUX SYSTEMES DE GESTION DE BASES DE DONNEES

Une base de données est un ensemble structuré de données partagées entre plusieurs applications, dans lequel les structures de données exploitées représentent aussi bien les données du monde réel que les associations entre ces données.

Définition AFNOR :

> Une base de données est une structure de données permettant de recevoir, de stocker, et de fournir à la demande, des données à de multiples utilisateurs indépendants.

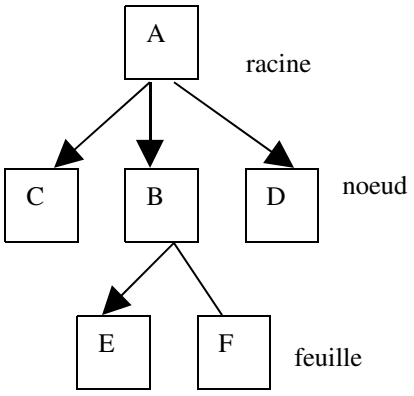
Avec le développement des bases de données depuis plusieurs décennies, le groupe ANSI/X3/SPARC, groupe de normalisation, a établi 3 ***niveaux de description*** des données :

- *le niveau externe*
  ou niveau utilisateur, est la description d'une partie de la base de données, correspondant à la vision d'une application particulière
- *le niveau conceptuel*
  correspond à une structuration sémantique des données du monde réel ; sans prise en compte des contraintes d'implantation sur une machine
- *le niveau interne*
  correspond à la structure de stockage des données (organisation et mode d'accès physiques)

Ces 3 niveaux sont à compléter par le *niveau logique* qui est le prolongement du niveau conceptuel avec une prise en compte des contraintes imposées par le SGBD en terme d'organisation logique et d'accès logique au données

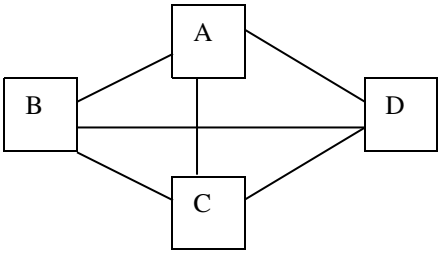
C'est à ce niveau logique que l'on prend en compte les ***3 grands modèles logiques*** exploités dans les bases de données :

- *le modèle hiérarchique (ou d'arbres)*
  ne permet de représenter qu'un seul type d'association : père-fils ; d'où sa limitation à la représentation d'univers hiérarchique. De plus le seul point d'accès est la racine (Exemple : IMS d'IBM).



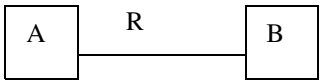
- *le modèle réseau*

permet la représentation de tous types d'associations mais impose pour accéder à un objet de naviguer le long de la base via une succession de pointeurs. De plus, il peut y avoir plusieurs liaisons entre les éléments, ou un élément ayant une ou des liaisons pointant sur lui même



- *le modèle relationnel*

  permet, en s'appuyant sur une théorie mathématique élaborée, de représenter de façon simple, sous forme de table aussi bien les objets du monde réel, que les associations entre ces objets.



Il existe une relation R entre A et B : R(A,B)

Un Système de Gestion de Bases de Données (SGBD) est un ensemble de composants logiciels permettant la mise en oeuvre de bases de données dans le respect d'un modèle (*dans le présent cours, le modèle relationnel*), et avec plusieurs objectifs, parmi lesquels :
- l'indépendance dans la définition des données du monde réel, par rapport au structures physiques de stockage
- le partage, qui peut être simultané, des données entre plusieurs applications
- l'indépendance logique des visions propres à chaque application
- la non redondance des données
- la cohérence des données

2.2. INTRODUCTION AU MODELE RELATIONNEL : quelques concepts

Le modèle relationnel respecte la règle selon laquelle tout modèle de données doit intégrer 3 composantes:
- des structures pour définir les données
- des opérateurs pour les manipuler
- des règles d'intégrité traduisant les contraintes liées à la définition des données par les structures du modèle

Le modèle a une double assise mathématique : il s'appuie à la fois sur la théorie ensembliste et sur la théorie des prédicats.

Principaux concepts du modèle relationnel :

➔Concept de RELATION et de DOMAINE

Définition mathématique ensembliste :

La *relation* est le sous-ensemble du produit cartésien de n ensembles.

$R \subset (D_1 \times D_2 \times D_3 \times \ \ldots \ \times D_n)$

Exemple : pour un entité AVION , la relation AVION est inclus dans

AVNUM x AVNOM x CAPACITE x LOCALISATION

Le modèle relationnel introduit une portée sémantique à cette définition mathématique, en introduisant la notion de *Domaine* de définition = ensemble de valeurs que peuvent prendre les propriétés caractérisant une entité.

La *définition* de la relation *au sens du modèle relationnel* est alors :
la *relation* est un sous ensemble du produit cartésien de n domaines

La relation est donc composée d'un ensemble de n-uplets, encore appelés *tuples*

Un prédicat est une expression contenant des variables qui devient une proposition quand on remplace ces variables par des valeurs.
Exemple :
« l'avion de numéro X est un Y de capacité égale à Z places dont le parking se trouve à W.
La relation peut être considérée comme une forme d'écriture concise du prédicat
Exemple :
Prédicat : « l'avion de numéro AVNUM et un AVNOM de capacité égale à CAPACITE places dant le parking se trouve à LOCALISATION »

Relation : AVION(AVNUM, AVNOM, CAPACITE, LOCALISATION)

Le même domaine pouvant apparaître plusieurs fois dans une relation, le concept d'*attribut* est introduit pour lever l'ambiguïté sur le rôle des propriétés concernées.
L'attribut explicite le rôle joué par un domaine dans une relation. Le *nom d'attribut* doit donc être unique dans une relation donnée.

➔ Concepts d'INTEGRITE

- Intégrité de domaine

- à la définition, contrôle de validité de l'attribut
- à la manipulation, contrôle des opérations; une comparaison d'attribut n'est acceptée que si les attributs appartiennent au même domaine

- Intégrité de relation

- au moins un attribut ou ensemble d'attributs particulier doit permettre d'identifier sans ambiguïté un élément donné de la relation, c'est la <u>*clé primaire*</u>. C'est donc le sous ensemble d'attributs d'une relation pour laquelle doit être vérifiée la règle d'unicité.

- Intégrité de référence

Dans tout ***schéma relationnel***, il existe 2 types de relations :

- les relations <u>indépendantes</u> ou statiques

  Exemples : pilote(numpil,nompil,prenpil,adrpil,datnaiss,tel, ville,salaire)
  avion (avnum, avnom, capacite, localisation)

- les relations <u>dépendantes</u> ou dynamiques dont l'existence des tuples dépend des valeurs d'attributs situés dans d'autre relations

  Exemple : vols (numvol,numpil,avnum,heurdep,duree,villdep,villarr)
  dépend de
  pilote(numpil,nompil,prenpil,adrpil,datnaiss,tel, ville,salaire)

La dépendance d'une relation est caractérisée par la présence de *clés étrangères*, i.e d'attributs qui sont clés primaires dans une autre relation du schéma. Cette dernière relation est qualifiée de relation de référence.

Exemple : NUMPIL et AVNUM dans VOLS

Dans ce cas, un certain nombre de contrôles doivent être faits pour garantir l'*intégrité référentielle* de la base de données :

- à l'insertion d'un tuple dans la relation dépendante, vérifier que les valeurs de clés étrangères existent dans les relations correspondantes
- à la demande de suppression d'un tuple dans une relation de référence, 4 possibilités :
  - interdire la suppression si la valeur de clé primaire est utilisée dans la relation dépendante
  - mettre à NULL (ou à une valeur par défaut) la valeur de la clé étrangère correspondante
  - informer l'utilisateur d'une incohérence éventuelle du schéma
  - supprimer les tuples utilisant la valeur concernée dans la relation dépendante

2.3. DEFINITIONS RELATIVES AUX BASES DE DONNEES RELATIONNELLES

La ***table*** est la visualisation à un instant donnée de la relation.

Exemple : Table ou Relation PILOTE

NUMEROPILOT	NOMPILOT	PREN_PILOT	VILLE	SALAIRE
004	TARTANPION	Albert	Paris	21 000,00
007	DUPONT	Jean	Lyon	19 000,00
010	WILSON	Fred	Marseille	18 000,00

une ***ligne*** représente un individu, une occurrence de l'entité représentée.
Une ***colonne*** représente un attribut de l'entité représentée.

A chaque attribut est associé un ***Domaine de Définition***.

Exemple :

NUMEROPILOT	Entier de 3 chiffres	①
NOM	Chaine de 25 caractères max	②
SALAIRE	Valeur numérique comportant 2 décimales, la partie entière comportant 6 chiffres max	③
VILLE	= PARIS, LYON, BORDEAUX, MARSEILLE,STRASBOURG,LILLES, RENNES	④

Dans les cas ①,② et③, il s'agit de définition par compréhension; dans le cas ④, d'une définition par extension.

Une ***CLE*** est un attribut ou ensemble d'attributs dont les valeurs sont systématiquement différentes pour chaque ligne. Il peut y avoir plusieurs clés. Ce sont des *Clés Candidates*. Pami elles, une seule sera désignée *Clé Primaire*, c'est celle qui, par convention, est retenue à titre principale comme clé de la relation. Les autres clés sont qualifiées de *Clés Secondaires*.

Le ***schéma d'une relation*** est la représentation de la relation sous la forme suivante :
Nom_Relation(Nom_Att_1**,** Nom_Att_2**, . .** Nom_Att_i**#,** Nom_Att_j**, . . ,**Nom_Att_n)

Où, l'attribut ou les attribut soulignés représente la clé primaire, les attributs suivis de **#** sont des clé étrangères.

Le ***schéma relationnel*** est constitué par l'ensemble des relations qui modélisent un monde réel.

Une ***VUE*** est une relation non matérialisée par une table. Elle correspondant à une vision utilisateur (niveau externe ANSI/X3/SPARC) à un instant donnée d'une ou plusieurs relations.

2.4 L'EXEMPLE D'ACCESS

SGBDR	Access
Schéma relationnel	Fichier base de données
Table	Table
Ligne	Ligne
Colonne	Champ
Clé primaire	Clé Primaire
Contraintes d'intégrité référentielle	Relations entre tables

3. INTERROGATION DES BASES DE DONNEES RELATIONNELLES

3.1. OPERATIONS ELEMENTAIRES SUR LES BASES DE DONNEES RELATIONNELLES

- *Projection* : sélection de l'ensemble des lignes de la table, en ne retenant qu'un sous-ensemble des attributs

Exemple : sur la table PILOTE(NUMPIL, NOMPIL, PRENPIL, ADRPIL, . . . ,SALAIRE)

NUMEROPILOT	NOMPILOT	PREN_PILOT
004	TARTANPION	Albert
007	DUPONT	Jean
010	WILSON	Fred

Notation : R1 = Proj(R,A_1,A_2,...A_p) où R est la relation source, et A_1,A_2,...A_p les attributs projetés

Exemple = Proj(PILOTE, NUMPIL, NOMPIL, PRENPIL)

Autre notation : R1 = $\Pi_{A^1,A^2,...A^p}(R)$

- *Restriction* : sélection d'un sous-ensemble de lignes répondant à une qualification donnée

Exemple : Pilote dont le salaire est strictement supérieur à 18000Francs
= restrict(pilote, salaire > 18000)

NUMEROPILOT	NOMPIL	PRENPIL	- - - - - - -	SALAIRE
004	TARTANPION	Albert	- - - - - - - -	21 000,00
007	DUPONT	Jean	- - - - - - - -	19 000,00

Notation : R1 = Restrict(R,C) où R est la relation source, et C le critère de qualification

Autre notation : R1 = $\sigma_C(R)$

- *Jointure* : entre 2 relations R et S de schéma différent, c'est l'ensemble des lignes de R et S satisfaisant à une condition portant sur au moins un attribut de chacune des relations.

exemple : entre PILOTE et VOLS, Nom des pilotes en service au départ de Nice = Nom des pilotes pour lesquels NUMEROPILOT dans PILOTE = NUMEROPILOT dans VOLS avec VILL_DEP=Nice.

Notation : R = Join(R1,R2,C) où R1 et R2 sont les relations sources, et C le critère de jointure

Autre notation : R = R1 $\bowtie_C$ R2

- *Union* : somme de 2 relations de ***même*** schéma, sur l'ensemble ou un sous-ensemble d'attributs.

Exemple : l'UNION de la vue Avionnice (= sélection sur Avion des avions localisés à Nice) et de la vue Avionairbus (= sélection sur Avion des avions de nom AirbusA300) donne l'ensemble des avions localisés à Nice et l'ensemble des avions A300

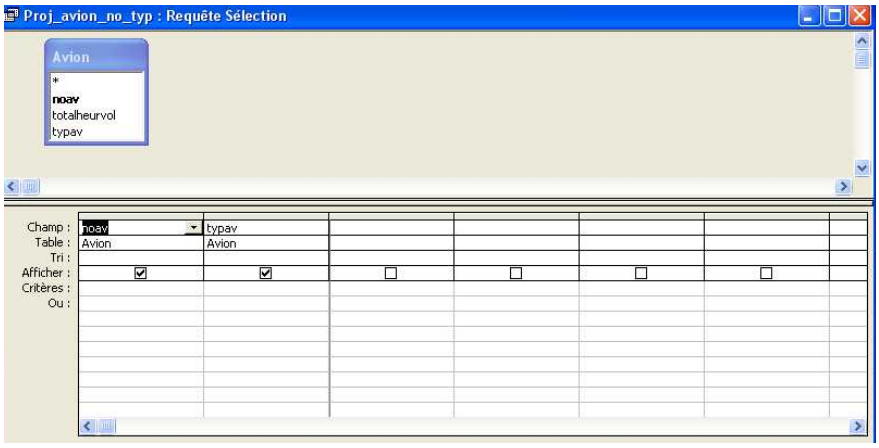
Notation : R1 = Union(R,S) où R et S sont les relations sources

L'algèbre relationnelle définit d'autres opérateurs non étudiés dans cet enseignement (différence, intersection, division, …)

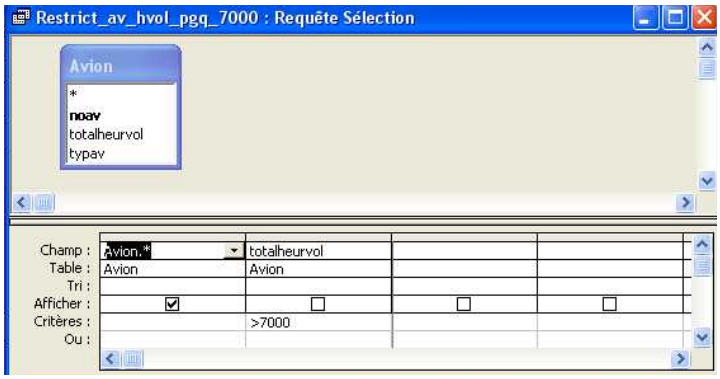
3.2. Mise en œuvre graphique sous MS- ACCESS du language d'interrogation

Mode création par défaut = mode graphique.

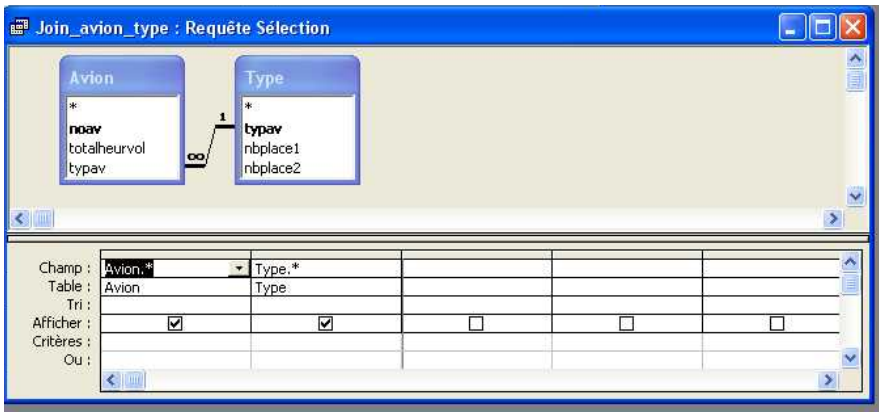
Projection



Restriction



Jointure



3.3. LE LANGAGE SQL : le sous-langage d'interrogation

SQL : **S**tructured **Q**uery **L**anguage
Expression proche de la langue naturelle anglaise des opérations souhaitées.
Comporte 2 sous langages : LMD (Language de Manipulation des Données) et LDD (Language de Définition des Données).

Seul le LMD sera étudié dans le cadre de ce cours.

Le LMD comporte le sous-langage d'Interrogation et le sous langage de mise à jour.

Le sous-langage d'interrogation est constitué d'un ***ordre*** (ou instruction) unique : ***SELECT***.

- La projection en SQL

  SELECT *A1, A2, A3, ….,An* FROM *nomtable* ;
- La restriction en SQL

  SELECT * FROM *nomtable* WHERE *expression logique*;
- La jointure en SQL

  SQL1 : SELECT * FROM *nomtable1, nomtable2* WHERE *critère de jointure*;

  SQL2 : SELECT * FROM *nomtable1* JOIN *nomtable2* ON *critère de jointure*;
- L'union en SQL

  SELECT ..... UNION SELECT ….;
- Les combinaisons d'opérations en SQL. ***Format générique*** de l'ordre SELECT traduisant une combinaison des opérations de projection, restriction et jointure

Format SQL1

SELECT [DISTINCT] <liste_d'attributs_projetés | *|expression(attribut)|fonction(attribut)>
FROM < nom_de_table [,nom_de_table …] >
[WHERE <critère_de_qualification et /ou critère_de_jointure>]
[ORDER BY < nom_attribut [, nom_attribut ...] <(ASC|DESC]] > ;

Format SQL2

SELECT [DISTINCT] <liste_d'attributs_projetés | *|expression(attribut)|fonction(attribut)>
FROM < nom_de_table [join nom_de_table [join …..on critère_de_jointure] on critère_de_jointure]
[WHERE <critère_de_qualification >]
[ORDER BY < nom_attribut [, nom_attribut ...] <(ASC|DESC]] > ;

Clause <u>DISTINCT</u> : permet d'obtenir une relation résultat dans laquelle tous les tuples sont différents (pas de doublons)

Clause <u>ORDER ... BY</u> : ordonne les tuples résultat selon les valeurs croissantes (ASC) ou décroissante (DESC) d'un ou plusieurs attributs

- Les opérateurs spécifiques SQL

IN : pour sélectionner un attribut dont la valeur est présente dans un ensemble donné de valeurs

Attribut IN (valeur1, valeur2, ….,valeurn)

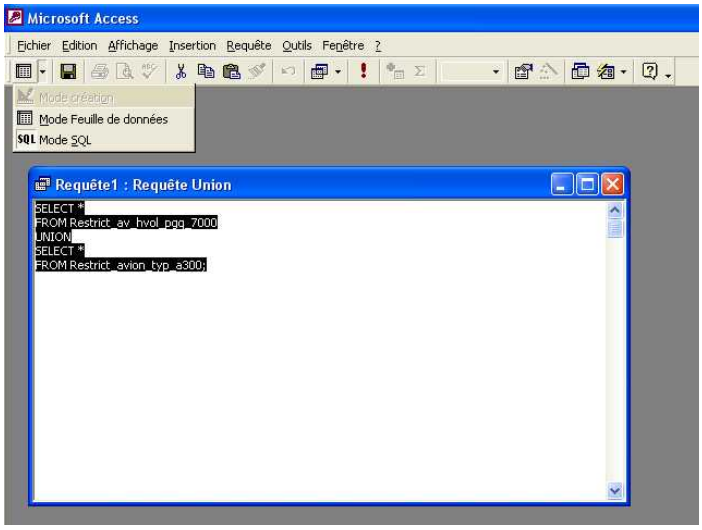
BETWEEN : pour sélectionner un attribut dont la valeur est comprise dans un intervalle donné de valeurs

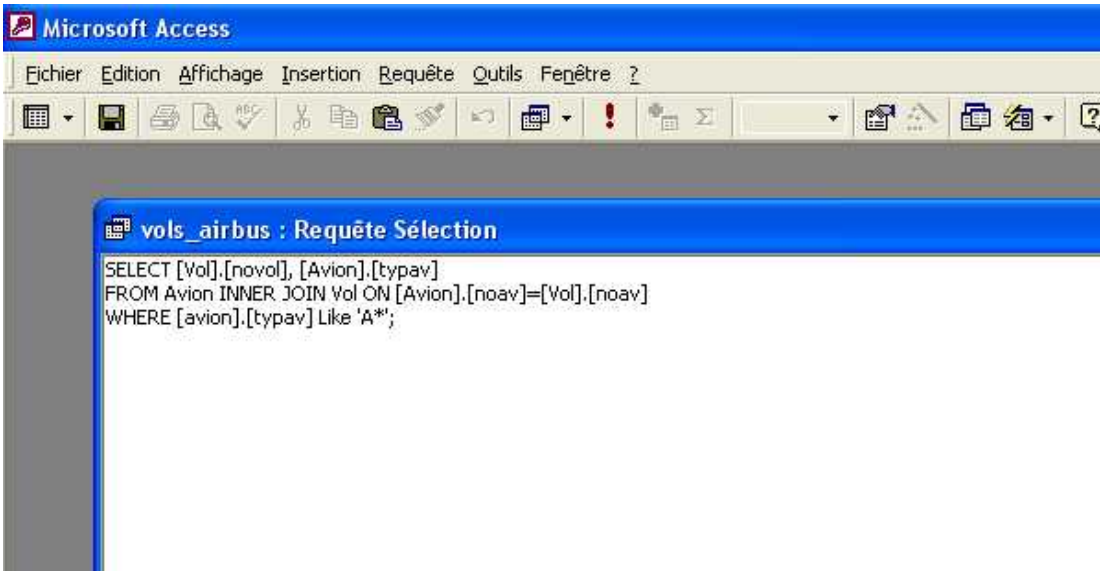
Attribut BETWEEN valeurDeb AND valeurFin

LIKE : pour comparer la valeur d'un attribut de type chaine de caractères à une chaine de caractères incluant des caractères génériques. '**%**' désigne une séquence de caractères de longueur variable (éventuellement nulle) ; '_' désigne un caractère quelconque.

Attribut LIKE chaine générique

3.4. SQL sous MS- ACCESS





Expression de la jointure : INNER JOIN

3.5. Requêtes SQL imbriquées

➢ Les requêtes imbriquées simples avec '=', 'IN'

Le résultat du SELECT imbriqué est utilisé comme valeur de comparaison dans un critère de qualification.

SELECT •
FROM •
WHERE AttributX = (SELECT …….) ;
Dans ce cas le SELECT ramène IMPERATIVEMENT une seule valeur

SELECT •
FROM •
WHERE AttributX IN (SELECT …….) ;
Dans ce cas le SELECT peut ramener plusieurs valeurs

Plusieurs niveaux d'imbrication sont possibles.

3.6. Requêtes SQL de regroupement

➢ Fonctions additionnelles

Les fonctions de calcul sur ensemble permettent d'effectuer des calculs sur une colonne (un attribut). Ce sont les fonctions suivantes :

Sum(attribut) : calcule la somme des valeurs de l'attribut sur un ensemble de tuples
Avg(attribut) : calcule la moyenne des valeurs de l'attribut sur un ensemble de tuples
Max(attribut) : détermine le maximum des valeurs de l'attribut sur un ensemble de tuples
Min(attribut) : détermine le minimum des valeurs de l'attribut sur un ensemble de tuples
Count(attribut) : compte le nombre de valeurs différente de l'attribut sur un ensemble de tuples

➢ Agrégat

L'Agrégat est un opérateur additionnel qui effectue

① un partitionnement de la relation opérande R sur les valeurs identiques des attributs de regroupement : $\{a_i, a_j, ..\}$,
② l'application sur chacune des partitions d'une (ou plusieurs) fonction(s) de calcul sur un (ou plusieurs) attribut(s) donné(s) : $f_i(a_k)$
③ une projection finale sur les attributs de regroupement et les fonctions de calcul

Notation :

R_r = Agrégat(R, $\{a_i, a_j, ..\}$, $f_i(a_k)$ [,$f_j(a_h)$, ..])

Format générique du SELECT avec regroupement (ou agrégat) :

Format SQL1

SELECT < nom_attribut1 [, nom_attribut2 ...]> [, fonction_de_calcul(), ….]
FROM < nom_de_table [,nom_de_table …] >
[WHERE <critère_de_qualification et /ou critère_de_jointure>]
[**GROUP BY** < nom_attribut1 [, nom_attribut2 ...]] >
[HAVING <critère_de_qualification>]
[ORDER BY < nom_attribut [, nom_attribut ...]] > ;

Format SQL2

SELECT [DISTINCT] <liste_d'attributs_projetés | *|expression(attribut)|fonction(attribut)>
FROM < nom_de_table [join nom_de_table [join …..on critère_de_jointure] on critère_de_jointure]
[WHERE <critère_de_qualification >]
[**GROUP BY** < nom_attribut1 [, nom_attribut2 ...]] >
[HAVING <critère_de_qualification>]
[ORDER BY < nom_attribut [, nom_attribut ...] <(ASC|DESC]] > ;

Les attributs projetés (clause select) à l'exception des fonctions de calcul, doivent être les mêmes que ceux de la clause GROUP BY.
La clause HAVING, restriction appliquée au regroupement, n'existe qu'en présence d'une clause GROUP BY.

4. MANIPULATION DES BASES DE DONNEES RELATIONNELLES

4.1. SQL : le sous-langage de mise à jour des données

➢ Insertion de données : INSERT

INSERT INTO < nom_de_table >
VALUES <expression [, <expression> …]> **;**
Ou
INSERT INTO < nom_de_table >
< ordre_select > **;**

➢ Suppression de données : DELETE

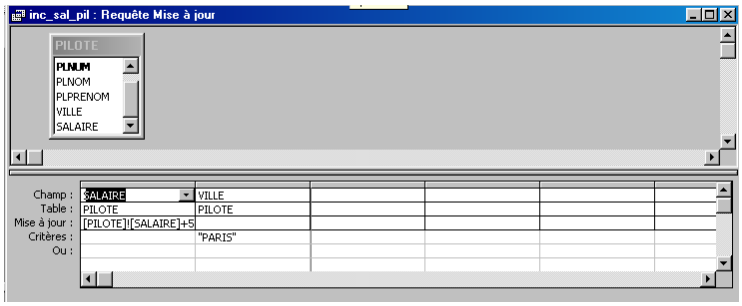
DELETE FROM <nom_de_table>
[WHERE < critère_de_qualification >]**;**

➢ Mise à jours des données : UPDATE

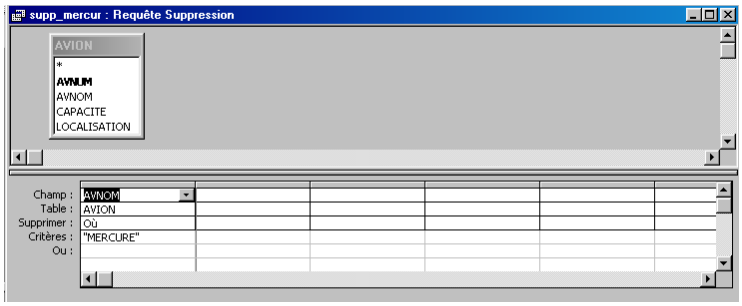
UPDATE < nom_de_table >
SET < nom_d'attribut_1 > = < expression1 | ordre_select >
[< nom_d'attribut_2 > = < expression2 | ordre_select > …]
[WHERE < critère_de_qualification >] **;**

4.2. Mise en œuvre graphique sous MS- ACCESS du langage de mise à jour

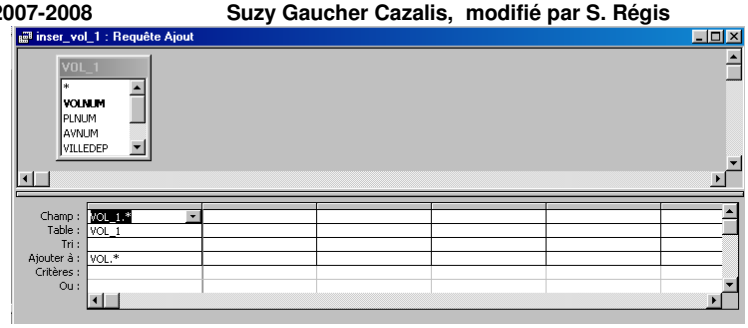
MISE A JOUR DES DONNEES



SUPPRESSION DE DONNEES



INSERTION



5. METHODE SIMPLIFIEE DE CONCEPTION D'UNE BASE DE DONNEES RELATIONNELLE

5.1 PROCEDURE D'ETABLISSEMENT DU SCHEMA RELATIONNEL

Conformément à l'architecture multi-niveau de l' ANSI/X3/SPARC, la conception d'une base de donnée passe par une première phase de modélisation conceptuelle (schéma conceptuel) avant d'aboutir au niveau logique au schéma relationnel.

Au niveau conceptuel, plusieurs démarches sont possibles :

- Méthodique : exploitation d'une méthode s'appuyant sur un modèle conceptuel de données (MERISE)
- Analytique : exploitation d'une grille d'analyse dans laquelle on recense, par une recherche systématique, toutes les données élémentaires. On regroupe ces données par entités logiques significatives de l'univers étudié, ce qui permet de recenser ensuite les association sémantiques entre ces entités.
- Intuitive : recensement de toutes les entités significatives et associations entre ces entités (évènements, entités permanentes). La descriptions de toutes ces entités et associations consiste à dresser pour chacune la liste exhaustive de ses propriétés.

La démarche proposée

La démarche proposée ici est analytique avec une justification permanente par l'intuitif.
On analyse la structure sémantique des données sans se soucier du mode d'implantation en machine.

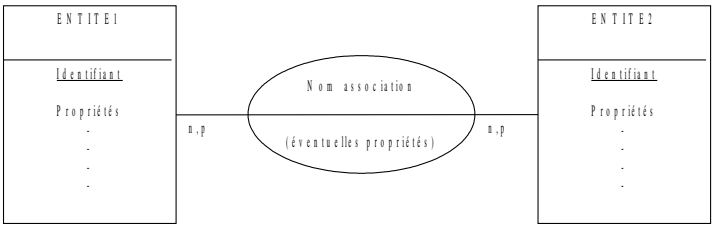
Les étapes d'établissement du schéma conceptuel des données sont les suivantes :

E1) Recenser dans le *Dictionnaire des données*, les **types de données élémentaires** (= plus petit élément d'information) qui représentent les propriétés des entités de l'univers réel (Ex . : nom d'un pilote, type d'un avion,….). C'est intuitivement que l'on repère les Entités.
Spécifier éventuellement les **règles** auxquelles sont soumises les données élémentaires(limite d'âge, valeur du kilométrage, …).

E2) Recenser les **types de données composées,** i.e. les **types d'entités** en regroupant les propriétés permettant de caractériser les entités du monde réel étudié (Ex. : étudiant, diplôme,…..). Pour cela, il est important, en premier lieu de repérer la propriété identifiante du type d'entité, c'est-à-dire celle dont la connaissance d'une valeur donnée permet d'identifier sans ambiguïté une et une seule occurrence d'entité donnée. Il est ensuite important de bien cibler les propriétés qui caractérisent vraiment une entité, i.e distinguer une propriété caractéristique d'une propriété associé (Ex. : « num_etudiant » caractérise le type d'entité « étudiant », alors que « type_inscription » est une propriété caractérisant l'inscription de l'étudiant à une UE et non l'étudiant lui-même). Une propriété caractéristique Px, d'une entité est une propriété telle que la connaissance d'une valeur donnée de la propriété identifiante Pi, permet de déduire une et une seule valeur de Px.

E3) Recenser les **associations** entre les entités spécifiées (Ex. : inscription qui associe l'entité « etudiant » à l'entité « unite_d_enseignement »). Une association est caractérisée au moins par les propriétés identifiantes des deux entités associées. Elle peut également posséder en plus des propriétés qui lui sont intrinsèques.

E4) Représentation du monde réel étudié par un « schéma conceptuel des données ». Nous utilisons ici des conventions de représentation inspirées de la méthode Merise, mais notons bien que *nous n'appliquons pas la méthode Merise*. Le formalisme est le suivant :



Chaque type d'entité est représenté par un rectangle donnant dans sa partie haute le nom de l'entité, et listant dans sa partie basse les propriétés de l'entité ; la propriété identifiante est soulignée.
Chaque association est représentée par un ovale traversé par un trait joignant les entités associées. De chaque coté du trait, on place une paire de nombre (min, max), appelée ***cardinalité***, qui expriment le nombre minimum et le nombre maximum d'occurrences de l'association pour une occurrence de chaque entité concernée.

Il s'agit maintenant à partir du schéma conceptuel, d'établir le schéma relationnel correspondant.
Les règles sont les suivantes :

R1) Chaque Entité est traduite par une Relation avec :
Clé primaire Relation =Identifiant Entité
{ attributs Relation } = {propriétés entité}

R2) Chaque Association sans Cardinalité (1,1) ou (0,1) avec ou sans propriétés intrinsèques, est traduite par une Relation avec :
Clé primaire *composite* Relation = { clé de Entité_1#, clé de Entité_2#, …} ; *clé de Entité_1 et clé de Entité_2 sont des clés étrangères, puisque référençant chacune la clé primaire d'une autre relation*
{ attributs Relation } = {propriétés intrinsèques association}

R3) Chaque Association avec au moins une Cardinalité (1,1) ou (0,1) se traduit par l'ajout d'une clé étrangère dans la Relation Dépendante (celle représentant l'entité avec les cardinalités (1,1) ou (0,1)) qui référence la clé primaire de la relation paire (dans l'association)

Structures support de la procédure

Dictionnaire des données élémentaires et recensement des entités :

Données élémentaires recensées	Nom abrégé	Type	Contrainte	Entités logiques
Numéro	NUMXX	Entier	>0 et <99999	YYYY
Nom	NOMXX	Ch. De 40 car alphabétiques		YYYY
Adresse	ADRXX	Ch. De 70 car alphanumériques		YYYY
----------	-----	--------	--------	----------
Axx	AXX	Réel		ZZZ
----------	-----	--------	--------	----------
----------	-----	--------	--------	----------

Recensement des associations entre entités :

Entités en association	XXXXX	YYYYYY	ZZZZZ
XXXXX		*Expression de l'association éventuelle*	*Expression de l'association éventuelle*
YYYYYY	*Expression de l'association éventuelle*		*Expression de l'association éventuelle*
ZZZZZ	*Expression de l'association éventuelle*	*Expression de l'association éventuelle*	

Schéma de relation :

YYYY (NUMXX, NOMXX,ADXX,REFXX#)

L'identifiant en gras est le nom de la relation

l'attribut souligné est la clé primaire (CONTRAINTE D'INTEGRITE DE RELATION)

l'attribut suivi de # est clé étrangère (CONTRAINTE D'INTEGRITE DE REFERENCE)

Normalisation du Schéma Relationnel

Pour éviter les redondances et faciliter les évolutions de la base de données, il faut s'assurer que les relations soient ***normalisées***, i.e conforme à 3FN.

1re Forme normale : 1FN

une relation est en 1FN si tous les attributs sont atomiques, i.e aucun attribut n'est lui-même décomposable sous forme de relation

la relation Client(code client, nom client, adresses) n'est pas en 1FN puisqu'il existe plusieurs adresses pour un même client

la relation Client(code client, nom client, adresse livraison, adresse facturation) est en 1FN

2e Forme normale : 2FN

une relation est en 2FN si elle est en 1FN et si tous les attributs autres que la clé dépendent de l'intégralité de la clé et pas d'une partie seulement

la relation Fournisseur(nom-fournisseur,code article, adresse, prix) n'est pas en 2FN puisque 'adresse' ne dépend que d'une partie de la clé, à savoir 'nom-fournisseur'

les relations Fournisseur(numéro-fournisseur, nom-fournisseur, adresse) et Produit(code article, numéro fournisseur, nom-article, prix) sont en 2FN

3e Forme normale : 3FN

une relation est en 3FN si elle est en 2FN et si aucun attribut non clé ne dépend d'un autre attribut non clé

la relation Voiture(numéro immat, marque, type, puissance,couleur) n'est pas en 3FN puisque 'type' attribut non clé permet de déterminer 'puissance'

les relations Voiture(numéro immat, type, couleur) et Modèle(type, marque, puissance) sont en 3FN

5.2. UN EXEMPLE SIMPLE

La gestion du personnel d'une petite entreprise dans laquelle

- un employé est identifié par un N°, et caractérisé par ses nom, prénom, date de naissance, grade et salaire. Le salaire n'étant pas une fonction directe du grade.
- un service est identifié par un N° et caractérisé par un nom

La gestion du personnel consiste à gérer

- l'organisation des services : connaissance du directeur , de la secrétaire qui sont des employés
- les affectations : type d'affectation caractérisé par un pourcentage de temps de l'employé dans le service

1) Dictionnaire des données élémentaires :

Données élémentaires recensées	Nom abrégé	Type	Contrainte	Entités logiques
Numéro d'un employé	NumEmp	Entier	>0 et <99999	
Nom d'un employé	NomEmp	Car(50)		
Date de naissance d'un employé	DatNaiss	Date		
Grade d'un employé	Grade	Entier		
Salaire d'un employé	Salair	Entier)		

Numéro de service	NumServ	Entier	>0 et <999	
Nom de service	NomServ	Alphanum(70)		
Type d'affectation à un service	TypAffect	Entier	>=25 et <=100	

2) Recensement entités :

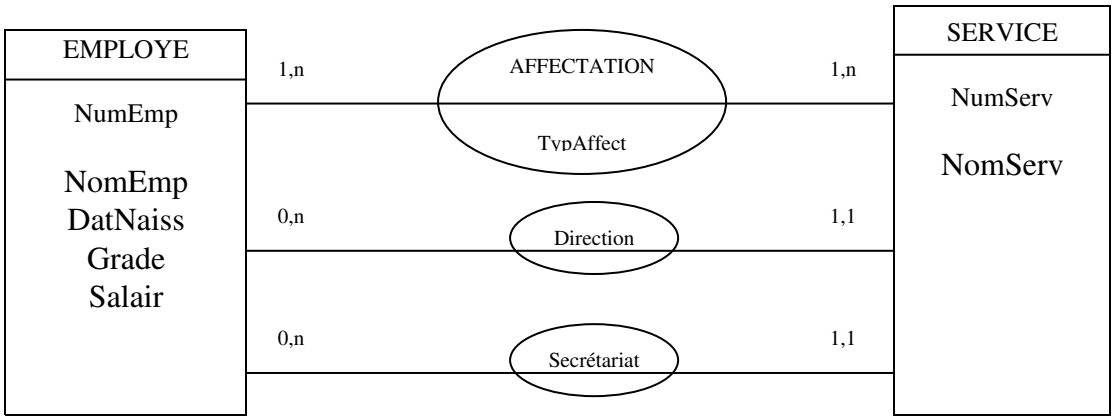
Données élémentaires recensées	Nom abrégé	Type	Contrainte	Entités logiques
Numéro d'un employé	NumEmp	Entier	>0 et <99999	EMPLOYE
Nom d'un employé	NomEmp	Car(50)		EMPLOYE
Date de naissance d'un employé	DatNaiss	Date		EMPLOYE
Grade d'un employé	Grade	Entier		EMPLOYE
Salaire d'un employé	Salair	Entier)		EMPLOYE
Numéro de service	NumServ	Entier	>0 et <999	SERVICE
Nom de service	NomServ	Alphanum(70)		SERVICE
Type d'affectation à un service	TypAffect	Entier	>=25 et <=100	

3) Recensement des associations

Entités en association	EMPLOYE	SERVICE
EMPLOYE		*Est Affecté à* **TypAffect** *% au service* ---------- *Est directeur du service* ---------- *Est Secrétaire du service*
SERVICE	*Utilise l'employé à* **TypAffect** *%* ---------- *A l'employé pour directeur* ---------- *A l'employé pour secrétaire*	

NOTE : plusieurs associations existent entre SERVICE et EMPLOYE
L'association d'affectation possède une propriété intrinsèque (TypAffect).

4) Schéma conceptuel des données



Détermination des cardinalités :
Pour 1 occurrence de EMPLOYE, il existe au minimum 1, et au maximum un nombre indéfini (n) d'occurrence de l'association AFFECTATION.
Pour 1 occurrence de SERVICE, il existe au minimum 1, et au maximum un nombre indéfini (n) d'occurrence de l'association AFFECTATION.
Pour 1 occurrence de EMPLOYE, il existe au minimum 0, et au maximum un nombre indéfini (n) d'occurrence de l'association Direction.
Pour 1 occurrence de SERVICE, il existe au minimum 1, et au maximum 1 occurrence de l'association Direction.
Pour 1 occurrence de EMPLOYE, il existe au minimum 0, et au maximum un nombre indéfini (n) d'occurrence de l'association Secrétariat.
Pour 1 occurrence de SERVICE, il existe au minimum 1, et au maximum 1 occurrence de l'association Secrétariat.

5) Schéma relationnel

Les entités **EMPLOYE** et **SERVICE** sont traduite chacune par une relation.
L'association **AFFECTATION** est traduite par une relation.
Les associations « direction » et « secrétariat » se traduisent par l'ajout d'une clé étrangère dans la relation SERVICE. Le directeur et la secrétaire étant des employés de l'entreprise, la relation de référence est la relation EMPLOYE.

Le schéma relationnel final est le suivant :

EMPLOYE (NumEmp, NomEmp, DatNaiss,Grade, Salair)

SERVICE (NumServ, NomServ, Directeur#, Secrétaire#)

AFFECTATION (NumEmp#, NumServ#, TypAffect)

Où Directeur est le numéro de l'employé dirigeant le service, et Secrétaire le numéro de l'employé assurant le secrétariat ; donc « directeur » référence NumEmp dans EMPLOYE, et « secretaire » référence également NumEmp dans EMPLOYE.

6) Normalisation du schéma relationnel

Il est en 1FN, puisque tous les attributs sont atomiques.

Il est également en 2FN, puisque dans l'unique relation à clé composite, AFFECTATION, le seul attribut non clé dépend de l'intégralité de la clé et pas d'une partie seulement.

Il est aussi en 3FN, puisque dans aucune des relations, il n'y a un attribut non clé qui dépend d'un autre attribut non clé.