

Outils Informatiques pour les Sciences

Module 5. Base de données *Partie 1 : modélisation*

Dr. Etienne Rivière
etienne.riviere@unine.ch
<http://iiun.unine.ch/>

Assistante : Maria Carpen-Amarie



Objectifs de ce module

- Introduire les principes des **bases de données relationnelles**
- Apprendre à **modéliser** une représentation du réel sous forme de relations avec le modèle *entité-association*
- Apprendre à interroger une base de données avec les **opérateurs relationnels du langage SQL**

Organisation de la séance 1

- Introduction à la modélisation d'une base de données, exemple de la base '*Observation de la faune*'
- Principes des bases de données et modélisation
- **Séance 2** : interrogation d'une base de données

Un monde de données

- Sur **Internet** : Allociné, IMDB, Wikipedia
- Dans **l'entreprise**
 - Gestion des stocks, des factures, des fournisseurs
 - Suivi de l'activité : location de voitures, vente de billets de spectacle, ...
- À **l'université**
 - Suivi d'une bibliothèque : gestion des emprunts
 - Gestion d'inscriptions, emplois du temps
- Gestion d'expériences dans un **laboratoire**
 - Gestion d'une animalerie, d'un stock de produits
- Observations lors d'expériences sur **le terrain**
 - Observations d'hygrométrie ou de météo
 - Inventaire et gestion de la flore dans une serre
 - **Comptage de population de la faune**

Exemple : observation de la faune

- Des **observateurs** se déplacent sur des **sites** du canton et **comptent** les **animaux** observés
 - Chaque observateur remplit des **fiches d'observation**

- Les **données** contenues dans les fiches servent à estimer
 - La répartition des aires d'observation pour chaque espèce
 - Le temps passé par chaque observateur sur le terrain
 - Les cohabitations observées entre espèces
 - La nature des sites où une espèce n'est jamais observée
 - Les périodes de l'année où chaque espèce est la plus observée

... et beaucoup d'autres interrogations et requêtes

Fiche individuelle d'observation

Date : 1 / 11 / 2013

Durée (minutes) : 135 minutes

Observateur
 Nom : Sandoz
 Prénom : Christine
 Fonction : Assistante-étudiante
 email : christine.sandoz@unine.ch

Site
 Nom du site : Lac des Taillères
 Localité : La Brévine
 Type (cocher une seule) : Forêt Marais Plaine Lac Rivière

Détail des animaux observés sur le site

Animal	Quantité
<i>Chat sauvage</i>	<i>3</i>
<i>Ecureuil</i>	<i>1</i>
<i>Renard</i>	<i>2</i>
<i>Loir</i>	<i>1</i>
Total	7

Fiche individuelle d'observation

Date : 22 / 10 / 2013

Durée (minutes) : 68 minutes

Observateur
 Nom : L'Eplatennier
 Prénom : Marie
 Fonction : Doctorante
 email : marie.lp@gmail.com

Site
 Nom du site : Les Planchettes
 Localité : La Chaux-de-Fond
 Type (cocher une seule) : Forêt Marais Plaine Lac Rivière

Détail des animaux observés sur le site

Animal	Quantité
<i>Blaireau</i>	<i>1</i>
<i>Sanglier</i>	<i>2</i>
<i>Loir</i>	<i>1</i>
<i>Ecureuil</i>	<i>2</i>
Total	6

Quel outil ?

- Comment **stocker** et **interroger** les données d'observation par des moyens informatique ?

	Pérennité	Facilité de collecte et de consultation	Recherches complexes
Scanner les fiches	✓	✗	✗
Recopier dans un tableau Excel	✓	✓	✗
Utiliser une base de données	✓	✓	✓

La base de données

« Observation de la faune »

- Objectif
 - **Collecter** les données facilement
 - Assurer qu'elles soient **stockées de façon pérenne**
 - **Extraire des informations** à l'aide de requêtes

- **Trois étapes** fondamentales
 - Décider du **modèle de stockage** des données
(la **structure** ou **schéma** de la base de données)
 - **Entrer les données** dans la base de données
 - **Interroger** la base de données avec des **requêtes**

- *Nous allons d'abord structurer la base de manière empirique, puis introduire les concepts de modélisation de bases de données en utilisant le même exemple*

Structurer les données

Programme d'observation de la faune dans le canton de Neuchâtel

Fiche individuelle d'observation

Date : 1 / 11 / 2013 Durée (minutes) : 135 minutes

Observateur

Nom : Sandoz
 Prénom : Christine
 Fonction : Assistante-étudiante
 email : christine.sandoz@unine.ch

Site

Nom du site : Lac des Taillères
 Localité : La Brévine
 Type (cocher une seule) : Forêt Marais Plaine Lac Rivière

Détail des animaux observés sur le site

Animal	Quantité
<u>Chat sauvage</u>	<u>3</u>
<u>Ecureuil</u>	<u>1</u>
<u>Renard</u>	<u>2</u>
<u>Loir</u>	<u>1</u>
Total	7

Données de l'observateur

Données du lieu

Données de l'observation

Données d'un détail

Division en tables

observateurs			
nom	prenom	fonction	email
Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch

lieux		
nom	localite	type
Lac des taillères	La Brévine	Lac

observations		
date	duree	total
1/11/2013	135	7

(peut être recalculé)

details	
animal	quantite
Chat sauvage	3
Ecureuil	1
Renard	2
Loir	1

- **Report « naïf »** des données dans des tables
 - Première observation -
 Le **total d'animaux** observés lors d'une séance d'observation peut être recalculé d'après les détails : il est **inutile de le conserver**

Structuration en table : la nécessité d'associations entre les données

observateurs			
nom	prenom	fonction	email
Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch
L'Eplatennier	Marie	doctorant	marie.lp@gmail.com

lieux		
nom	localite	type
Lac des taillères	La Brévine	Lac
Les Planchettes	La Chaux-de-Fond	Foret

observations	
date	duree
1/11/2013	135
22/10/2013	68

details	
animal	quantite
Chat sauvage	3
Ecureuil	1
Renard	2
Loir	1
Blaireau	1
Sanglier	2
Loir	1
Ecureuil	2

- Ajoutons les données de notre **deuxième fiche** ...
 - **Il est impossible de reconstituer les fiches d'observation d'origine !**
 - Quel est le lieu d'une observation, qui a fait cette observation, à quelle observation correspond un détail donné ?

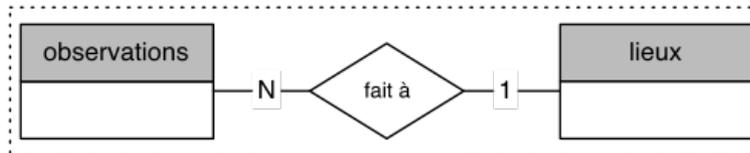
Association entre *observations* et *lieux*

observateurs			
nom	prenom	fonction	email
Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch
L'Eplatennier	Marie	doctorant	marie.lp@gmail.com

lieux			
<i>id_lieu</i>	nom	localite	type
1	Lac des taillères	La Brévine	Lac
2	Les Planchettes	La Chaux-de-Fond	Foret

observations		
<i>id_lieu</i>	date	duree
1	1/11/2013	135
2	22/10/2013	68
2	4/1/2014	15

details	
animal	quantite
Chat sauvage	3
Ecureuil	1
Renard	2
Loir	1
Blaireau	1
Sanglier	2
Loir	1
Ecureuil	2



- **Association entre données** : chaque ligne de la table *observations* doit être liée à exactement une ligne de la table *lieux*
 - Une ligne de la table *lieux* est associée à N lignes de la table *observations* ...
... mais 1 ligne de la table *observations* est liée à une seule ligne de la table *lieux*
 - On identifie une **relation $N \rightarrow 1$ (many-to-one)**
- ***id_lieu*** identifie de façon **unique** chaque ligne de la table *lieux*

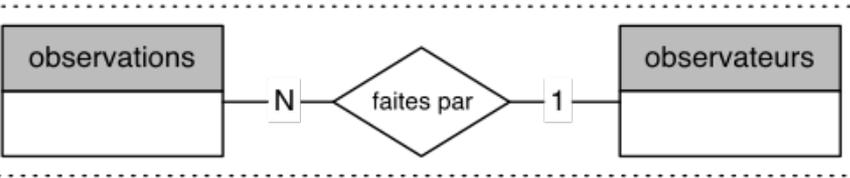
Association entre *observations* et *observateurs*

observateurs				
<i>id_personne</i>	nom	prenom	fonction	email
1	Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch
2	L'Eplatennier	Marie	doctorant	marie.lp@gmail.com

lieux			
<i>id_lieu</i>	nom	localite	type
1	Lac des taillères	La Brévine	Lac
2	Les Planchettes	La Chaux-de-Fond	Foret

observations			
<i>id_personne</i>	<i>id_lieu</i>	date	duree
1	1	1/11/2013	135
2	2	22/10/2013	68
1	2	4/1/2014	15

details	
animal	quantite
Chat sauvage	3
Ecureuil	1
Renard	2
Loir	1
Blaireau	1
Sanglier	2
Loir	1
Ecureuil	2



- On applique le même principe :
 - Une ligne de la table *observations* pointe vers une seule ligne de la table *observateurs*
 - Une ligne de la table *observateurs* peut être pointée par plusieurs lignes de la table *observations*
 - Relation $N \rightarrow 1$ (*many-to-one*)
- id_personne* identifie de façon unique chaque ligne de la table *observateurs*

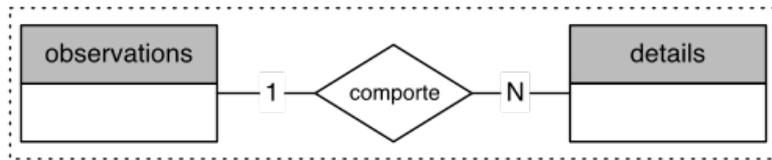
Association entre *details* et *observations*

observateurs				
<i>id_personne</i>	nom	prenom	fonction	email
1	Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch
2	L'Eplatennier	Marie	doctorant	marie.lp@gmail.com

lieux			
<i>id_lieu</i>	nom	localite	type
1	Lac des taillères	La Brévine	Lac
2	Les Planchettes	La Chaux-de-Fond	Foret

details		
<i>id_observation</i>	animal	quantite
1	Chat sauvage	3
1	Ecureuil	1
1	Renard	2
1	Loir	1
2	Blaireau	1
2	Sanglier	2
2	Loir	1
2	Ecureuil	2

observations				
<i>id_observation</i>	<i>id_personne</i>	<i>id_lieu</i>	date	duree
1	1	1	1/11/2013	135
2	2	2	22/10/2013	68



- Association **1 → N** entre la table *observations* et la table *details*
 - Une ligne de *observation* est pointée par plusieurs lignes de *details* (ou aucune !)
 - Une ligne de *details* pointe vers une seule ligne de *observations*
- **id_observation** identifie de façon unique chaque ligne de *observations*

Attention aux redondances !

observateurs				
<i>id_personne</i>	nom	prenom	fonction	email
1	Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch
2	L'Eplatennier	Marie	doctorant	marie.lp@gmail.com

lieux			
<i>id_lieu</i>	nom	localite	type
1	Lac des taillères	La Brévine	Lac
2	Les Planchettes	La Chaux-de-Fond	Foret

observations				
<i>id_observation</i>	<i>id_personne</i>	<i>id_lieu</i>	date	duree
1	1	1	1/11/2013	135
2	2	2	22/10/2013	68

details		
<i>id_observation</i>	animal	quantite
1	Chat sauvage	3
1	Ecureuil	1
1	Renard	2
1	Loir	1
2	Blaireau	1
2	Sanglier	2
2	Loir	1
2	Ecureuil	2

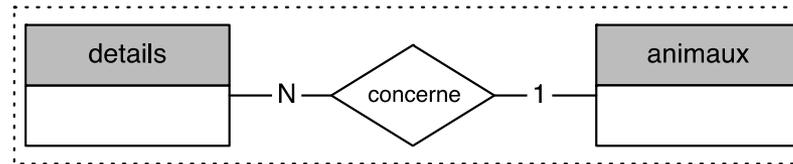
- **Redondance** : la même information stockée plusieurs fois
 - Nom des animaux « **Loir** » et « **Écureuil** » stockés de manière redondante
- Imaginons que l'on veuille **ajouter des informations** pour chaque type d'animal : protégé ou non, famille, régime alimentaire, ...
 - Pour modifier un animal, **on doit modifier plusieurs lignes de la table *details***
 - Si on modifie une partie des lignes pour un animal ⇒ **incohérence**
- Pour éviter les incohérences, on applique la règle : **une « entité » = une seule ligne dans une seule table**, et on utilise une association

Éliminer les redondances de la table *details*

details		
<i>id_observation</i>	<i>id_animal</i>	quantite
1	1	3
1	2	1
1	3	2
1	4	1
2	5	1
2	6	2
2	4	1
2	2	2

animaux		
<i>id_animal</i>	nom	protege
1	Chat sauvage	VRAI
2	Ecureuil	FAUX
3	Renard	FAUX
4	Loir	FAUX
5	Blaireau	VRAI
6	Sanglier	FAUX

(ajout de détails supplémentaires facile)



- **Nouvelle table *animaux*** destinée à conserver de manière unique les noms des animaux observés
- Si on modifie le nom « Ecureuil » par « Ecureuil gris » dans la table *animaux*, la **modification sera valable pour toutes les entrées associées** à la ligne avec *id_animal* = 2 dans la table *details*
- Si on veut ajouter des informations supplémentaires sur les animaux, on doit modifier seulement la table *animaux*
 - Par exemple, une colonne « protege » qui prend comme valeur *vrai* ou *faux*

Notre base de données complète

observateurs				
<i>id_personne</i>	nom	prenom	fonction	email
1	Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch
2	L'Eplatennier	Marie	doctorant	marie.lp@gmail.com

lieux			
<i>id_lieu</i>	nom	localite	type
1	Lac des taillères	La Brévine	Lac
2	Les Planchettes	La Chau-de-Fond	Foret

observations				
<i>id_observation</i>	<i>id_personne</i>	<i>id_lieu</i>	date	duree
1	1	1	1/11/2013	135
2	2	2	22/10/2013	68

details		
<i>id_observation</i>	<i>id_animal</i>	quantite
1	1	3
1	2	1
1	3	2
1	4	1
2	5	1
2	6	2
2	4	1
2	2	2

animaux		
<i>id_animal</i>	nom	protege
1	Chat sauvage	VRAI
2	Ecureuil	FAUX
3	Renard	FAUX
4	Loir	FAUX
5	Blaireau	VRAI
6	Sanglier	FAUX

Que peut-on faire avec cette base de données ?

- Stockage **pérenne** : le logiciel de gestion de base de données conserve les tables de données et la logique qui les lie (les **relations**) sur disque
- Les données peuvent être **interrogées**. Quelques exemples:
 - Quels sont les prénoms et noms des observateurs doctorants ?
 - `SELECT observateurs.prenom, observateurs.nom FROM observateurs WHERE fonction='doctorant'` ;
 - Quels sont les types de lieux où l'on a observé des chats sauvages?
 - `SELECT L.type FROM lieux L, observations O, details D, animaux A WHERE A.nom = 'Chat sauvage' AND A.id_animal=D.id_animal AND D.id_observation=O.id_observation AND O.id_lieu = L.id_lieu;`
 - Combien d'animaux protégés ont été observés en tout ?
 - `SELECT SUM(quantite) AS Total_protege FROM details WHERE id_animal in (SELECT id_animal FROM animaux WHERE protege=True);`
 - Ces requêtes sont en langage **SQL** ; pas besoin de les comprendre pour l'instant !

Les SGBD

- La gestion d'une base de donnée est un problème informatique complexe, pris en charge par un **logiciel spécialisé**
 - **SGBD** = système de gestion de base de données
- Rôles
 - **Organisation** des données : sous forme de tables permanentes sur le disque, avec des index pour assurer des accès rapides
 - **Gestion** des données : vérifier que les contraintes d'intégrité du schéma de la base de donnée soient respectées
 - **Accès** aux données : interface pour l'utilisateur ou l'administrateur mais aussi aux programmes faisant appel à la base de données
- Dans ce cours : **Access** (Microsoft)
(base de donnée mono-utilisateur)
 - Autres systèmes répandus : MySQL, SQL server (Microsoft), Oracle



Notions élémentaires

Schéma

observateurs				
<i>id_personne</i>	nom	prenom	fonction	email
1	Sandoz	Christine	assistant-étudiant	christine.sandoz@unine.ch
2	L'Eplatennier	Marie	doctorant	marie.lp@gmail.com
3	Robert	Charles	doctorant	
4	Decotterd	Stephane	professeur	stephane.d@bluewin.ch
5	Mitchum	Robert	assistant-étudiant	robert@mitchum.com
6	Tissot	Sophie	professeur	sophie.tissot@unine.ch
7	Bond	James	doctorant	007@mi5.gouv.uk
8	Zueger	Béat	professeur	

Données

Colonne obligatoire

Colonne facultative

Ligne (entité)

- Une base de donnée **relationnelle** est un ensemble de **tables**
 - Chaque table a un **nom unique** (exemple : *observateurs*)
 - Chaque table contient les **données relatives à des entités de même nature**
 - **Une ligne = une entité**
 - **Une colonne = une propriété** (ex. nom, fonction), avec un **type** (texte, nombre, date, ...)
- Noms de tables, noms des colonnes et domaines = **schéma** de la base de données
- Lignes (entités) = **contenu** de la base de données

Schéma d'une base de données

- Le **schéma** définit la structure de chacune des tables
 - **Nom** de la table
 - Pour chaque colonne
 - Le **nom** de la colonne
 - Son **type** (texte, nombre, date, choix vrai/faux, ...)
 - Son caractère **obligatoire** ou non
 - Des contraintes sur le contenu, ex. : accepte ou non un texte de longueur 0
 - La **clé primaire** et la ou les **clés étrangères**
- Le contenu des tables change fréquemment, le schéma est plus difficile à modifier
 - Il est important de bien faire la **modélisation** dans un premier temps

Types de colonnes (Access)

- Obligatoire ou facultative
- Chaque colonne contient un seul type de données
 - **Texte** : « Robert » ; « Hérisson des indes »
 - **Nombre**
 - **Entier** par défaut : 5154, -56, 0, 8998321
 - Nombre **flottant** (approximation d'un réel) : 3.1415, -10.9
- Choix **binaire** vrai/faux (True/False)

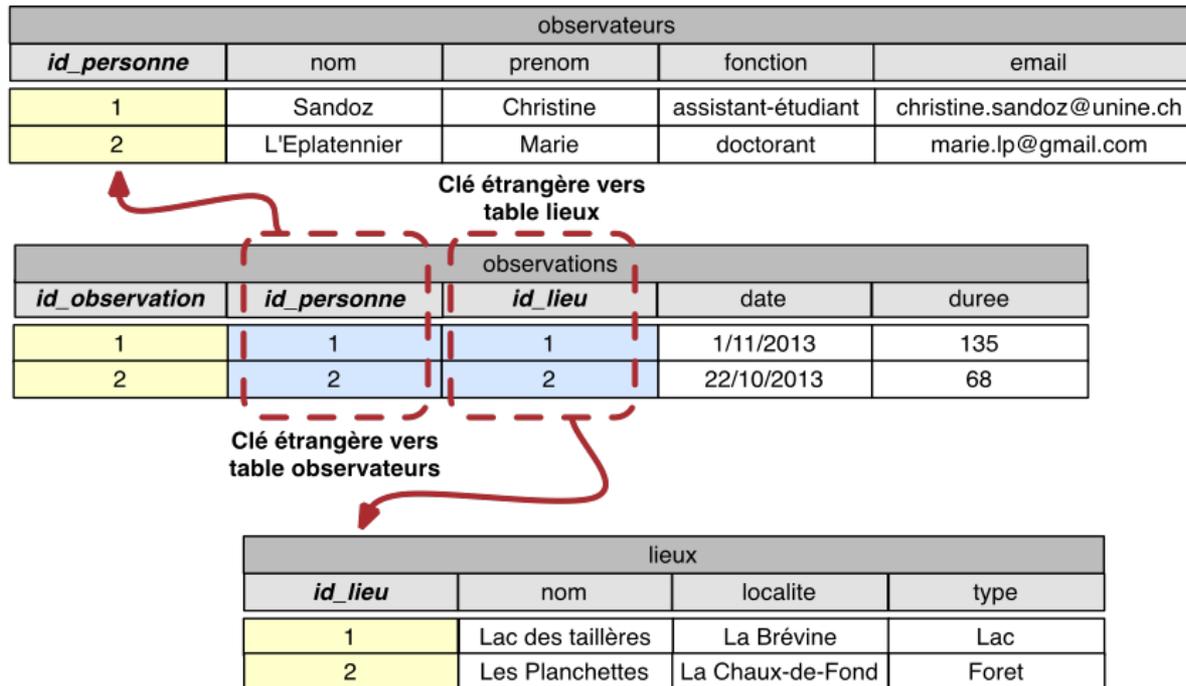
Clé primaire

animaux		
<i>id_animal</i>	nom	protege
1	Chat sauvage	VRAI
2	Ecureuil	FAUX
3	Renard	FAUX
4	Loir	FAUX
5	Blaireau	VRAI
6	Sanglier	FAUX

Clé primaire

- La **clé primaire** (ou **identifiant**) est **une colonne ou un groupe de colonnes** d'une table T tels que
 - À tout moment, il n'y a qu'**une seule ligne avec les mêmes valeurs pour ces colonnes** (deux lignes ne peuvent pas avoir la même valeur pour *id_animal*)
 - Seulement composé de **colonne(s) obligatoire(s)**
- La clé primaire permet de **désigner de façon unique une ligne de T**

Clés étrangères



- Une clé étrangère est une **colonne** ou un **groupe de colonnes** d'une table T tels que
 - Pour chaque ligne, il existe **dans une autre table S** une ligne dont la **clé primaire** a **pour valeurs** celles de cette colonne ou ce groupe de colonnes
 - La valeur de la clé étrangère dans T sert à **référencer une ligne de la table S**

Les clés peuvent être déclarées comme plusieurs colonnes

frais_deplacement			
<i>depart</i>	<i>arrivee</i>	kilometres	indemnite
Neuchatel	Bern	49.1	100 CHF
Neuchatel	Basel	123	150 CHF
Neuchatel	Zurich	151	175 CHF
Yverdon	Bern	75.9	150 CHF
Yverdon	Basel	168	180 CHF
Yverdon	Zurich	196	200 CHF

Clé primaire

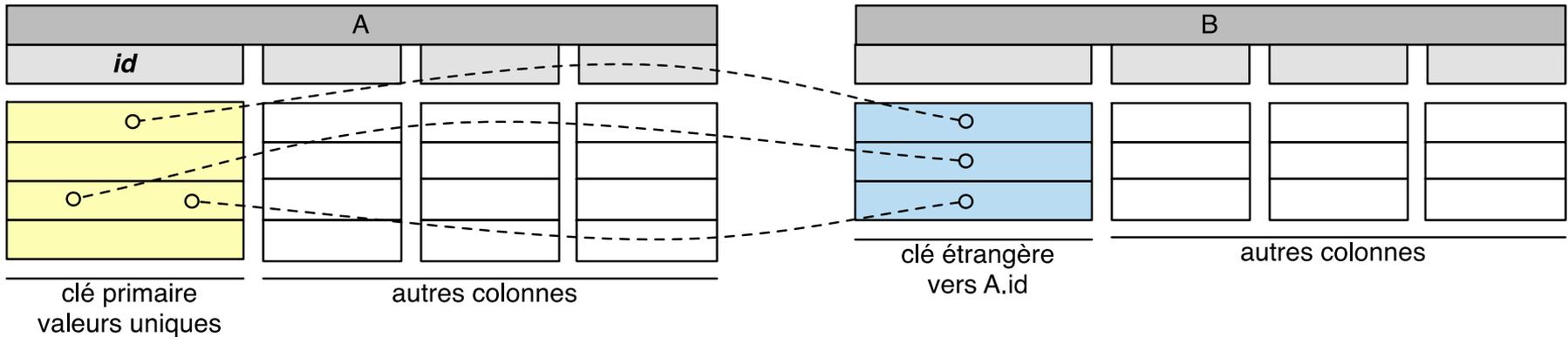
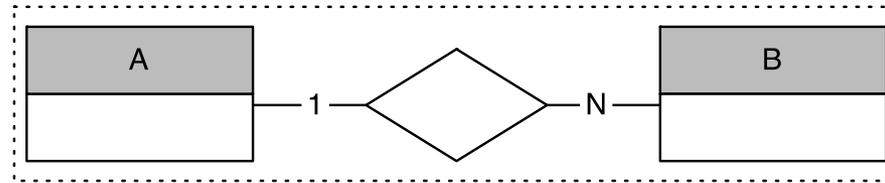
Clé étrangère vers table personnel

missions			
id_employe	depart	region	date
1	Neuchatel	Bern	14/11/2013
2	Yverdon	Bern	3/12/2013

Clé étrangère vers table frais_deplacement

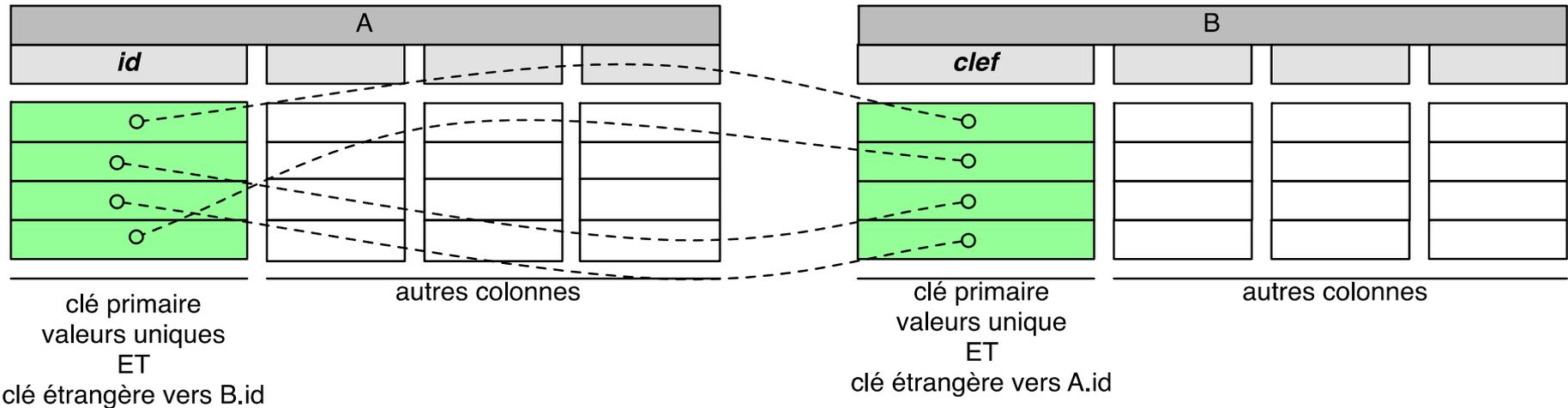
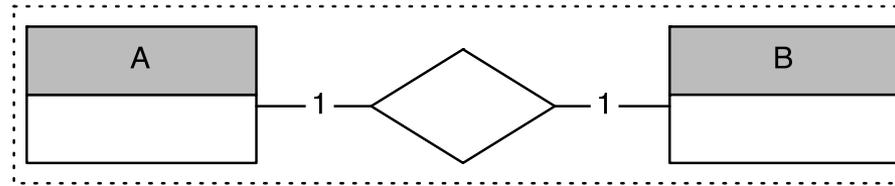
(table non représentée)

Traductions des associations en relations entre tables



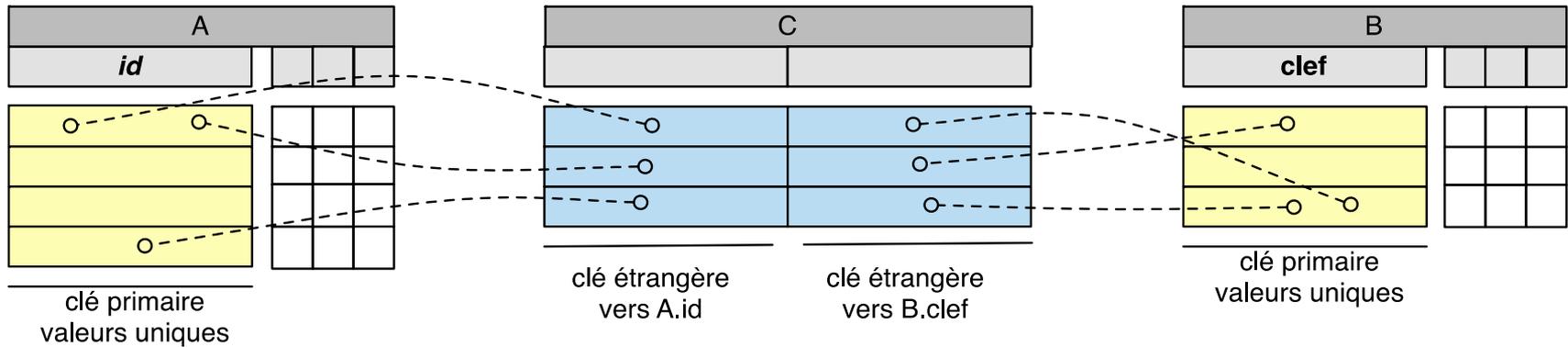
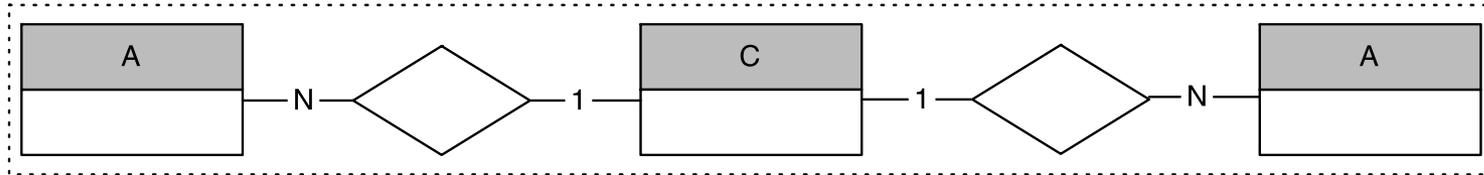
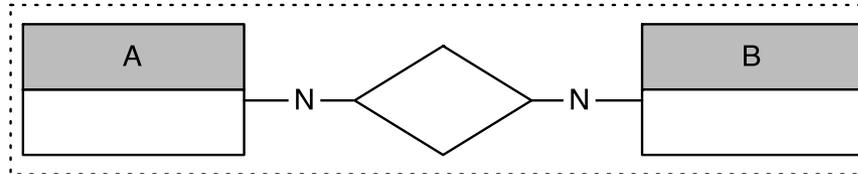
- **Relation 1 - N** entre table A et table B
 - La table A doit avoir une clé primaire (par ex., *id*)
 - La table B définit une clé étrangère vers A.id

Traductions des associations en relations entre tables



- **Relation 1 - 1** entre table A et table B
 - La table A doit avoir une clé primaire (par ex., *id*)
 - La table B doit avoir une clé primaire (par ex., *clef*)
 - A définit *id* comme clé étrangère vers B.clef
 - B définit *clef* comme clé étrangère vers A.id
- Il peut être plus simple de **fusionner** les deux tables !

Traductions des associations en relations entre tables

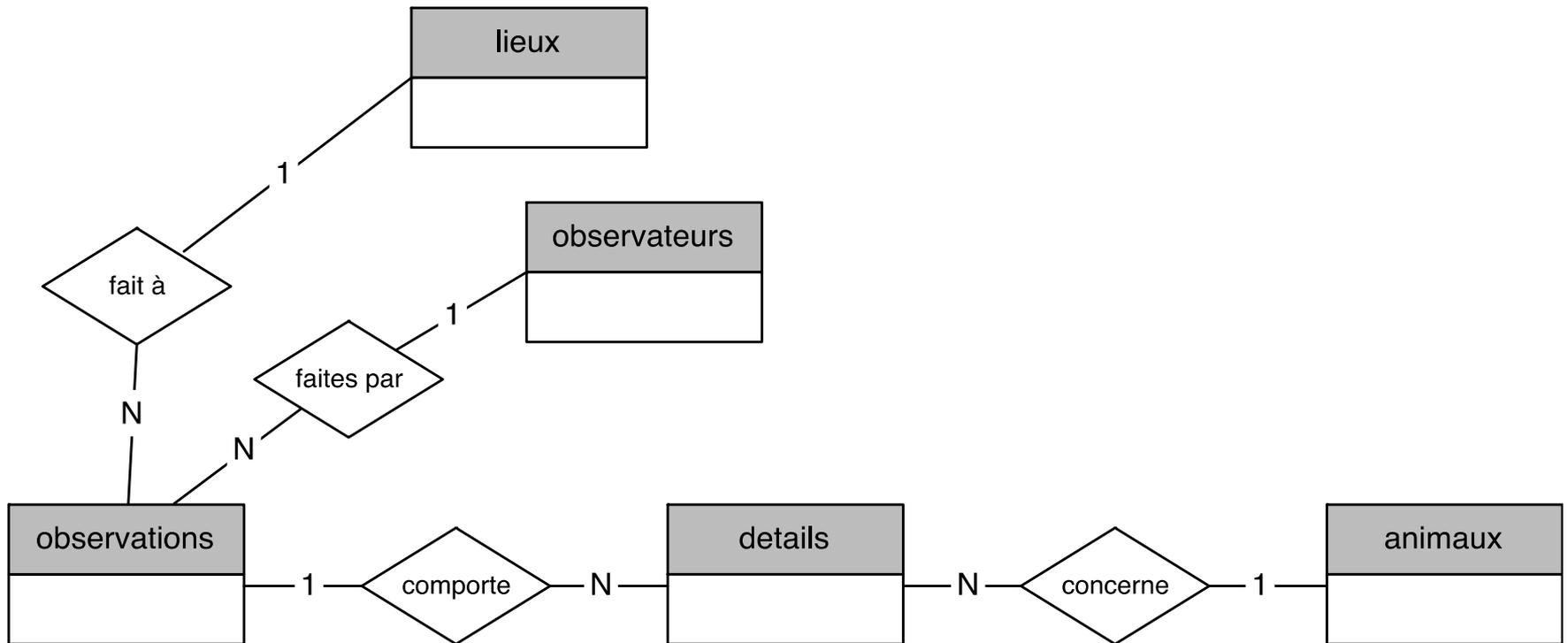


- **Relation N - N** entre la table A et la table B
 - Une ligne de A peut être liée depuis plusieurs lignes de B, et une ligne de B peut être liée depuis plusieurs lignes de A
 - Nécessite la création **d'une troisième table C**
 - A et B ont une clé primaire
 - C définit deux clés étrangères : une vers la clé primaire de A, une vers la clé primaire de B

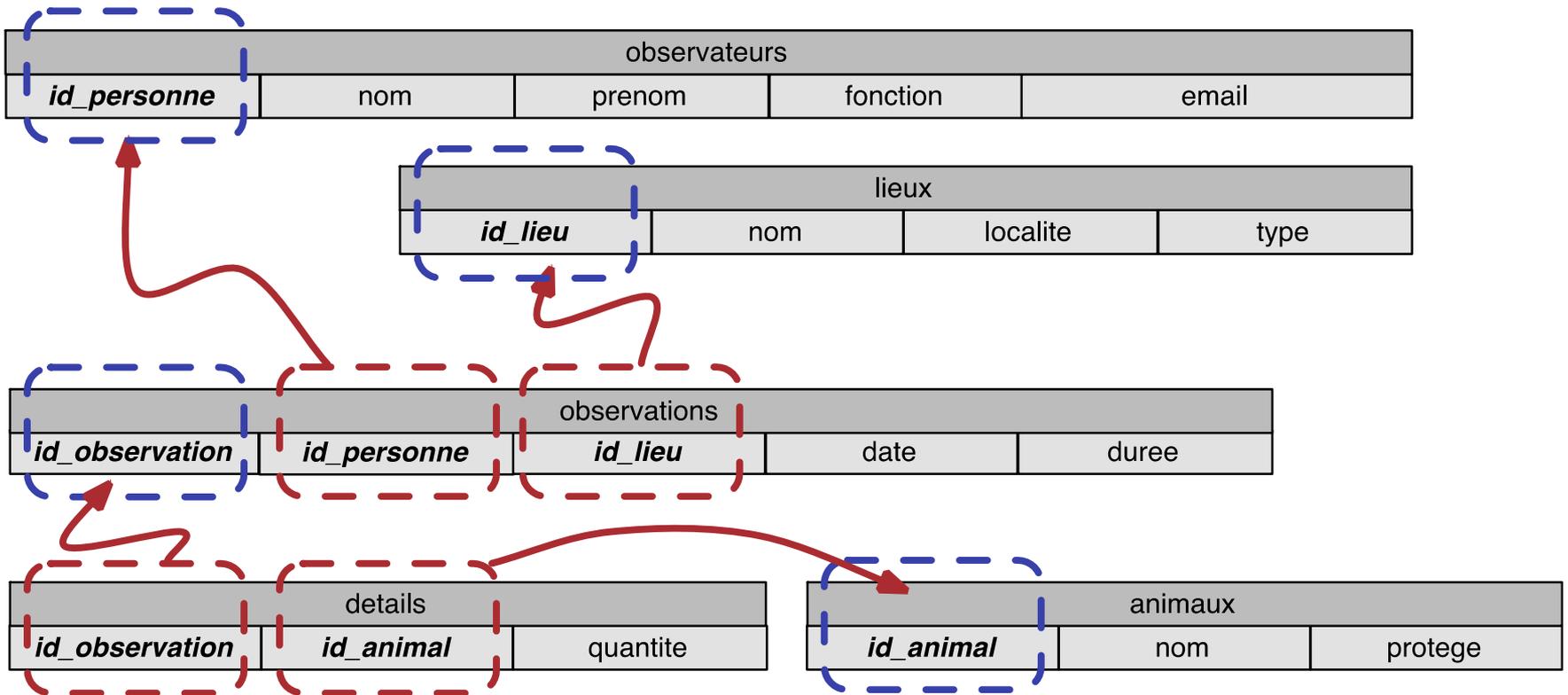
Contraintes référentielles

- **Contraintes** déclarées au SGBD
 - Découlent de notre **modélisation**
- Pour assurer **l'intégrité des données**
 - La création de nouvelles lignes ne sera pas autorisée si les contraintes d'intégrité ne sont pas respectées
- Contraintes de **deux types**
 - Définition de la clé primaire (**unicité**)
 - On ne pourra pas créer une nouvelle ligne avec des valeurs déjà utilisées pour la ou les colonnes de la clé primaire
 - Définition des clés étrangères (**existence**)
 - On ne pourra pas créer une nouvelle ligne, dont les valeurs pour les colonnes de la ou les clés étrangères n'apparaissent dans aucune ligne de la table liée

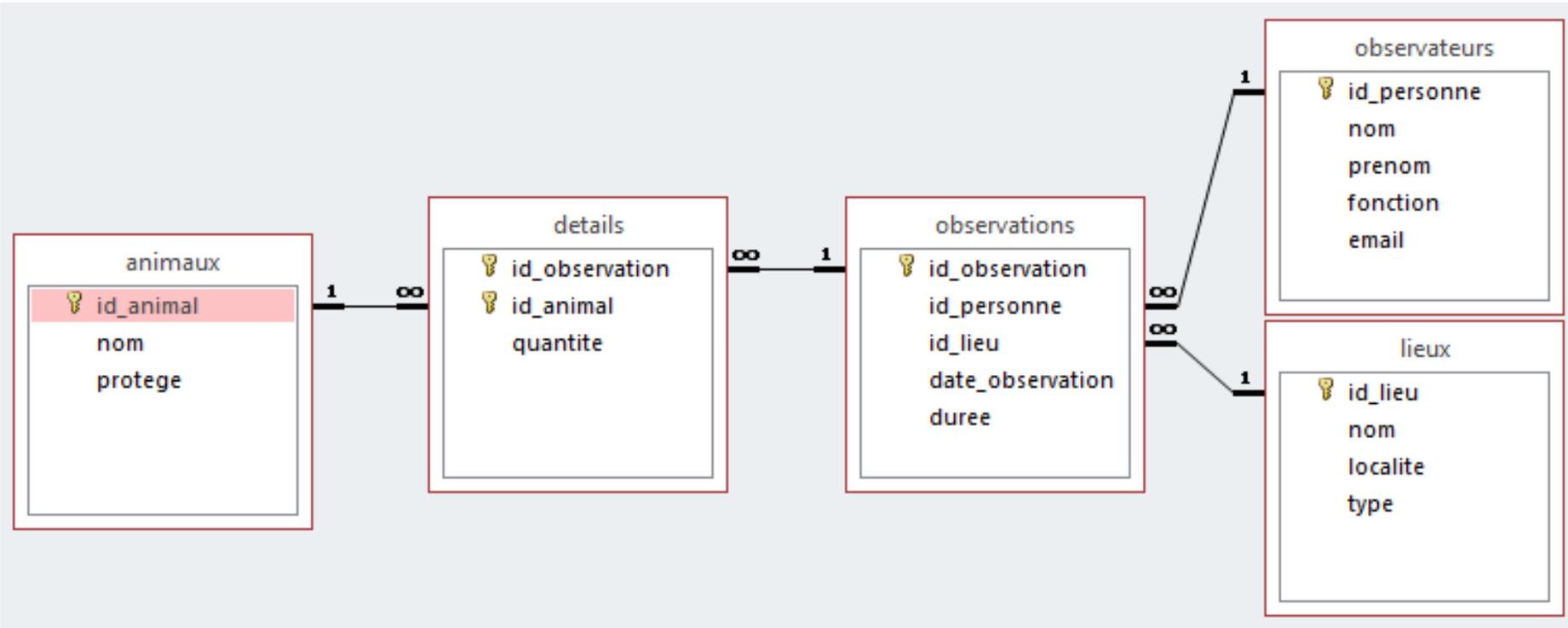
Modélisation



Contraintes référentielles



Contraintes référentielles (présentation Access)



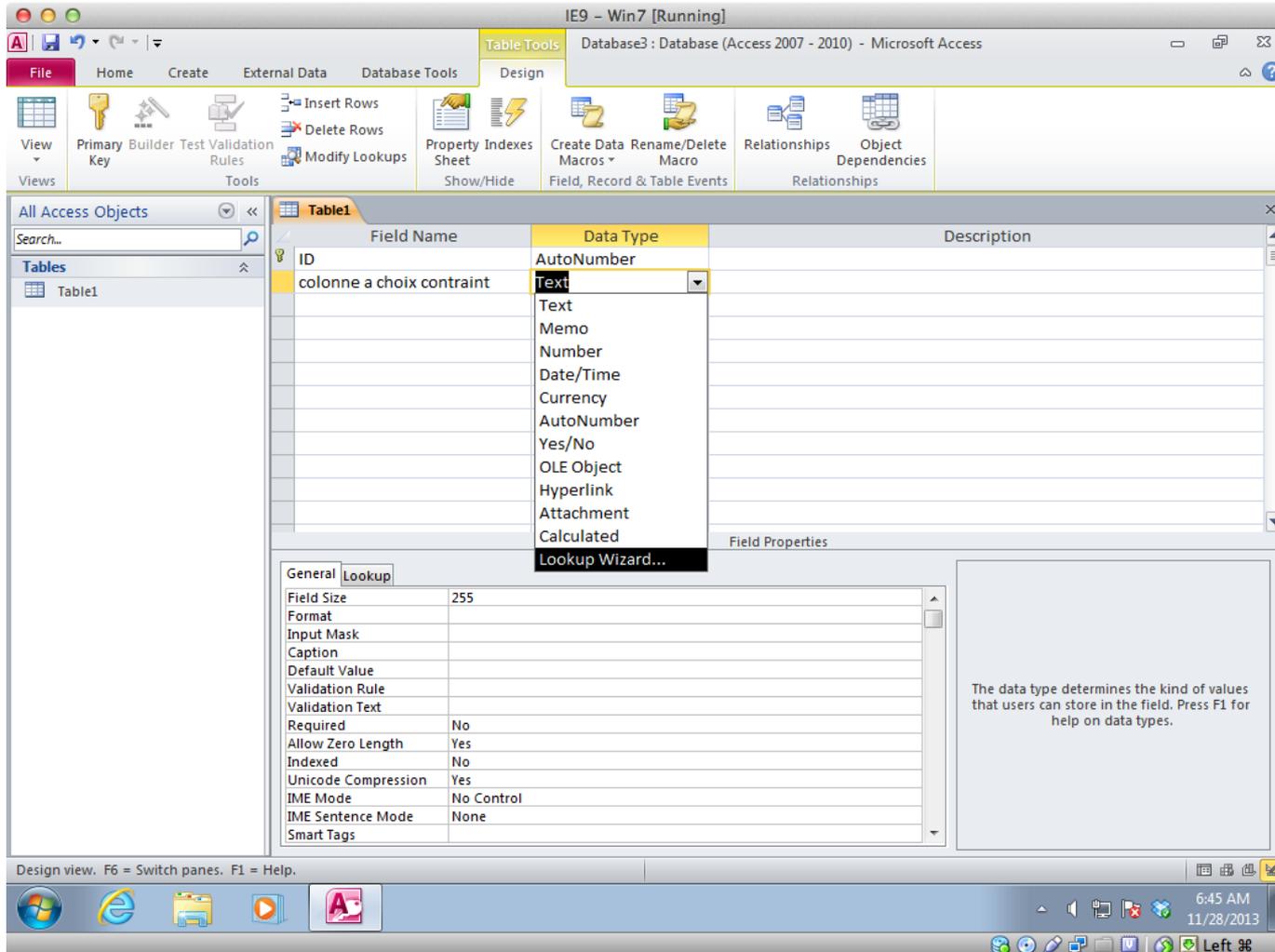
Tutoriel

- Création de la base de données « *observation des animaux* » sous Access
 - Définition des **tables**
 - Définition des **colonnes** et des **types**
- Spécification des contraintes d'intégrité
 - Définition des **clés primaires**
 - Définition des **clés étrangères**
 - Test du **respect des contraintes**

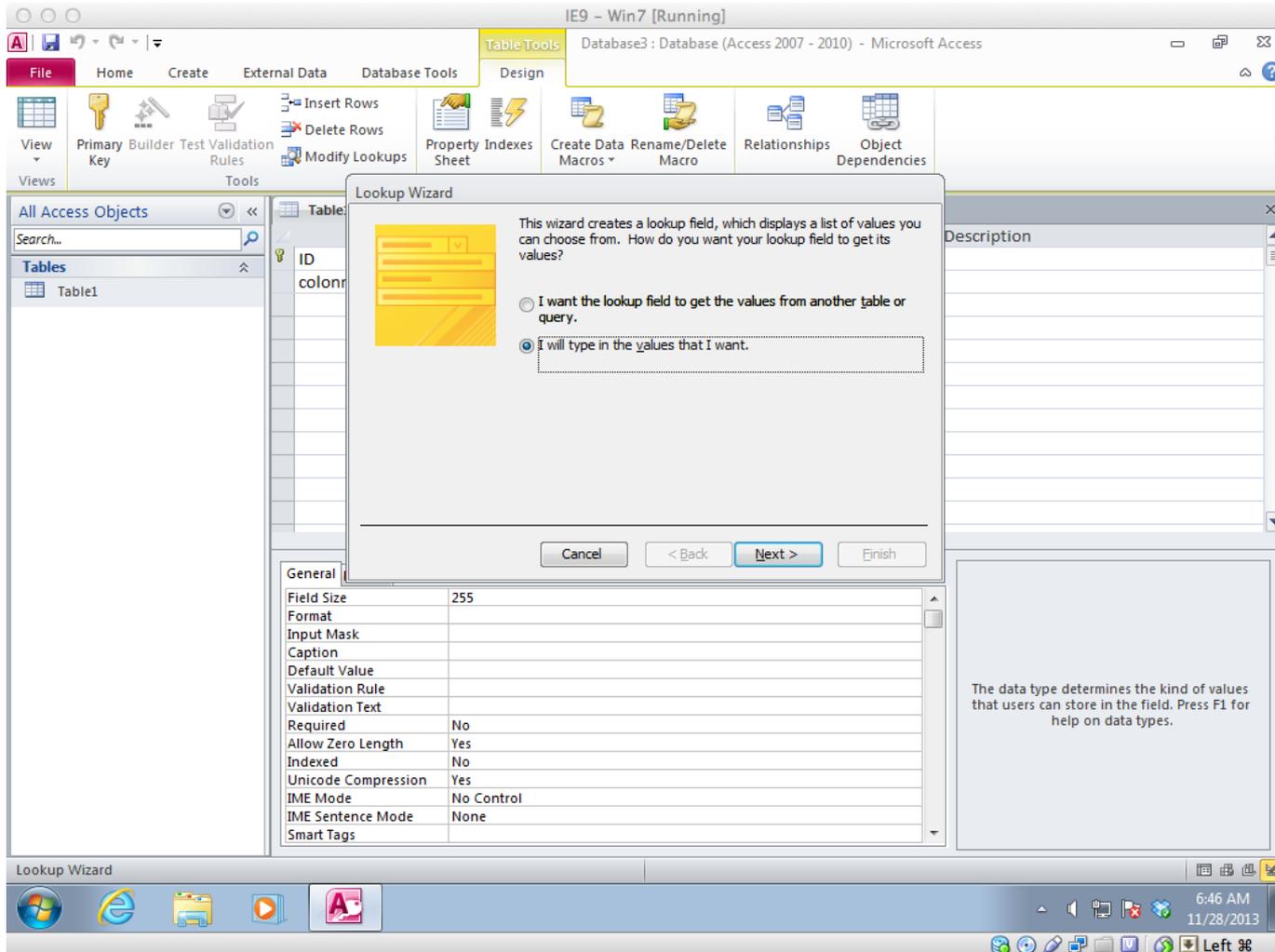
Définition d'un choix multiple

- Ouvrir la table en mode « design »
- Entrer le nom de la colonne
- Dans « data type », sélectionner « Lookup wizard »
 - Cocher « *I will type in the values that I want* »
 - Entrer les valeurs choisies dans la première colonne, puis Next
- On peut restreindre les possibilités à la liste en cochant « *Limit to list* »
- On valide l'opération en ouvrant la table en mode données

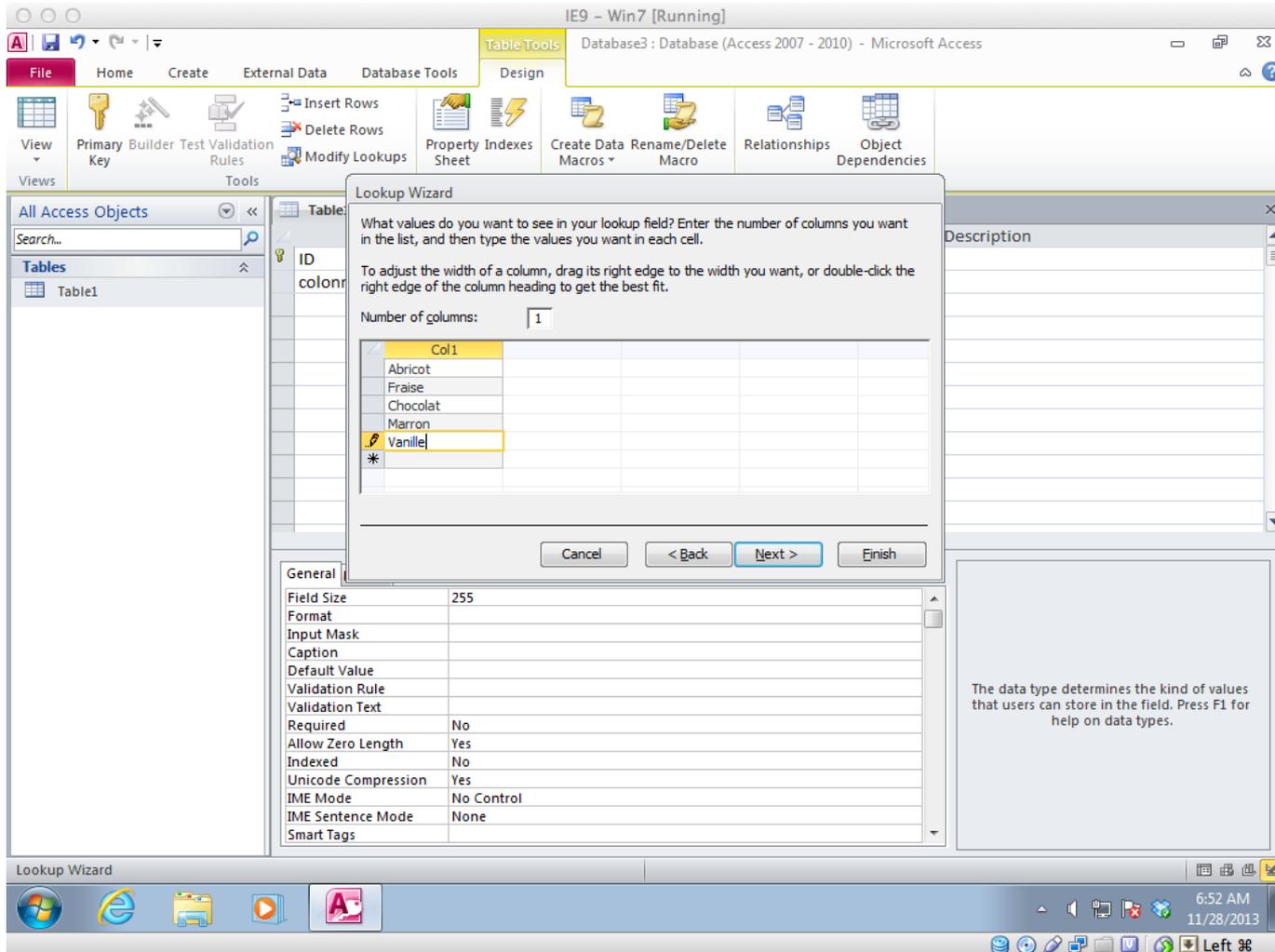
Définition d'un choix multiple 1



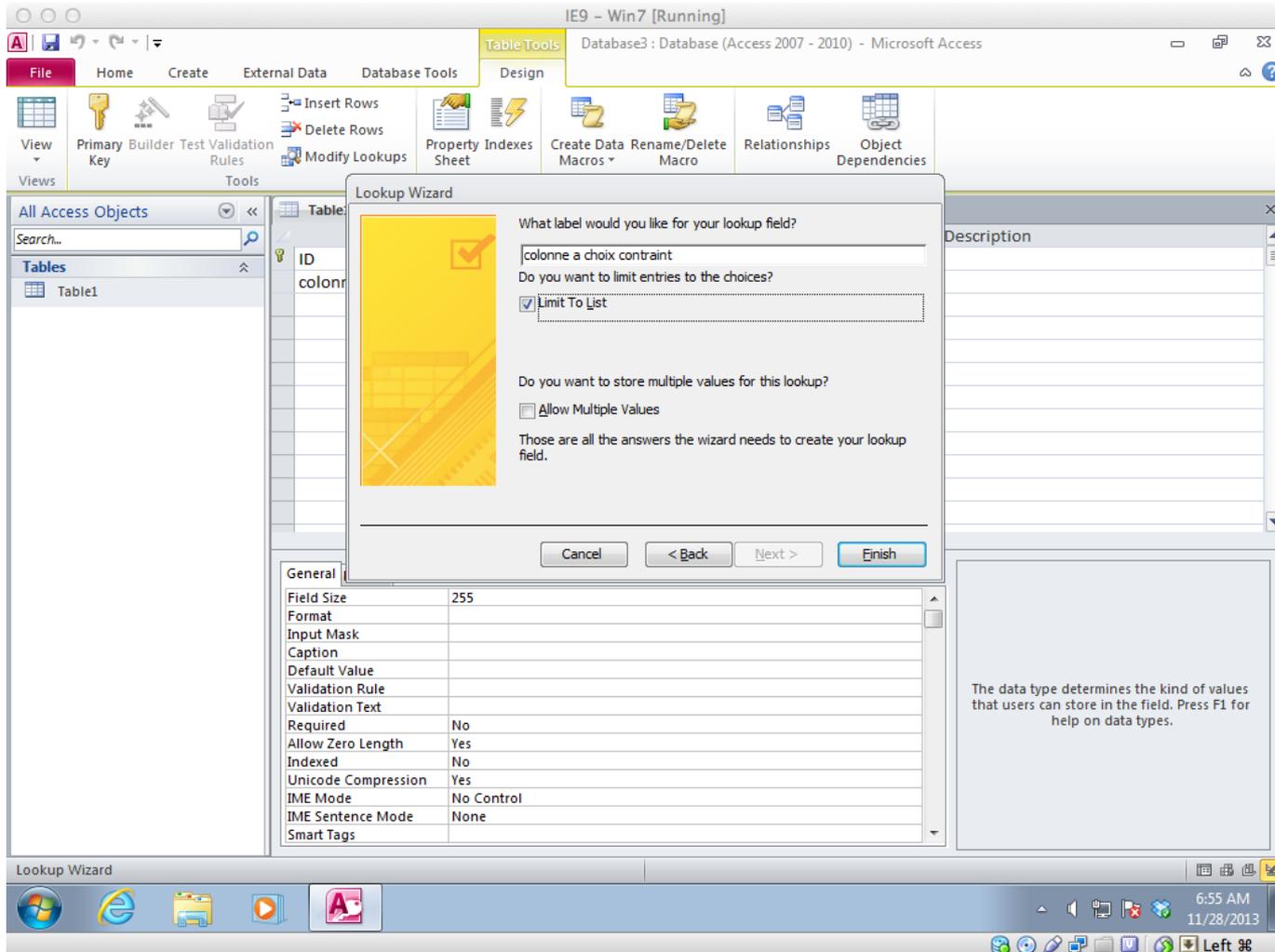
Définition d'un choix multiple 2



Définition d'un choix multiple 3



Définition d'un choix multiple 4



Exercice : base « cours »

- Objectifs
 - **Modéliser** la base de données selon cahier des charges
 - **Créer** la base de données sur Access
 - Mettre en place les **contraintes d'intégrité**
 - Entrer des **données de test** et vérifier qu'il n'est pas possible d'entrer des données incohérentes
- 3 versions
 - Gestion des inscriptions aux cours
 - + Gestion des notes
 - + Gestion des assistants
- Instructions de rendu sur la feuille d'exercice