

David TREMBLAY 1748125

Dominique PICHÉ 1766981

Laboratoire 3

Modélisation et modèle relationnel

INF3710

Bases de données

Groupe 02

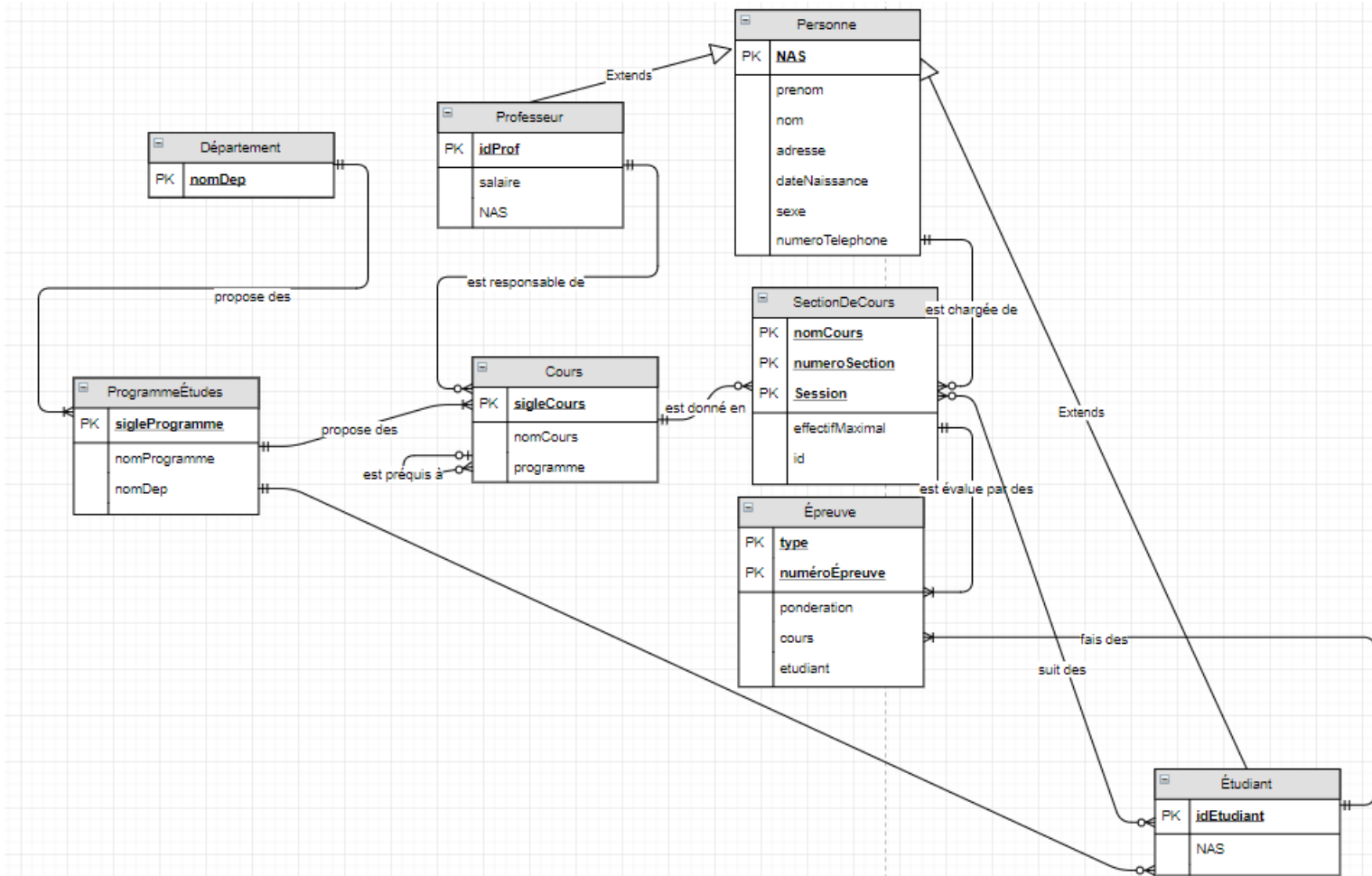
Département de génie informatique et génie logiciel

École polytechnique de Montréal

9 octobre 2017

Question a

1. Nous avons modélisé avec la crow's foot notation :



2.

Département: Chaque département doit posséder un nom unique. De plus, chacun de ces départements propose un ou plusieurs programmes d'études.

Programme d'études: Les programmes d'études sont identifiés par leur sigle unique distinctif (CIV, LOG, INF, etc.) ainsi que leur nom complet. Chaque programme d'étude propose plusieurs cours pour l'obtention du BAC dans ce programme. Il est à noter qu'au moins un programme doit exister pour qu'un département existe.

Cours: Chaque cours est identifié par un sigle unique où les trois premiers caractères permettent d'identifier le programme du cours et la suite permet d'identifier le cours. Le cours possède aussi un nom (exemple: INF3710 Bases de données). Certains cours peuvent être pré requis à d'autres cours. Cette situation est identifiée par la boucle sur cette entité. Au moins un cours doit exister pour chaque programme d'étude. Un cours

peut posséder aucune, une ou plusieurs sections de cours (aucune si le cours ne se donne pas pendant la session).

Personne: Une personne désigne un individu le plus général possible (sans aucun statut spécial). Tous les attributs sauf le numéro d'assurance social peuvent être identiques pour deux personnes différentes (deux personnes qui se nomment Dominique Piché par exemple). Nous utilisons donc le NAS qui est unique pour chaque individu comme clé primaire. Une personne peut être ou ne pas être chargée de sections de cours. Comme décrit par la suite, les entités professeur ainsi qu'étudiant héritent de personne et peuvent donc aussi être chargés de cours.

Professeur: Un professeur hérite d'une personne et a comme attribut supplémentaire son salaire. Nous l'identifions avec son identificateur de professeur et nous lui assignons un foreign key avec le NAS de l'entité personne puisque nous avons besoin de cette information. Finalement, un professeur peut n'être responsable d'aucun cours ou être responsable d'un ou plusieurs cours.

Étudiant: Semblable à l'entité professeur, l'entité étudiant hérite d'une personne. Il possède aussi son identificateur d'étudiant (matricule) ainsi qu'un foreign key qui est son NAS dans l'entité personne. Un étudiant doit compléter au moins une épreuve pour être évalué dans un cours mais il peut y avoir plus d'épreuves. Puisqu'un étudiant est une personne, il peut aussi être chargé de cours.

Épreuve: L'entité épreuve utilise une clé primaire combinée. La première clé primaire est le type d'épreuve (devoir, tp, contrôle, etc.) et la deuxième clé est le numéro de l'épreuve. On constate donc que ces deux clés peuvent être identiques dans plusieurs épreuves mais pas les deux clés en même temps (on ne peut pas avoir deux tp 1 par exemple). Nous avons aussi l'attribut pondération qui sert, comme son nom l'indique, à pondérer l'épreuve.

Section de cours: Comme l'entité précédente, cette entité utilise aussi une clé primaire combinée. Par contre cette clé primaire contient trois éléments et non deux. Les clés primaires nomDeCours et numeroSection représentent le cours et la section. La clé primaire session permet de savoir dans lesquelles des sessions le cours va être donné. On peut donc voir qu'une seule combinaison unique de ces trois éléments doit être possible. Il ne pourrait pas y avoir deux groupes 2 de INF3710 durant la même session par exemple. Finalement, l'attribut effectifMaximal indique le nombre d'étudiants pouvant s'inscrire dans la section et id attribut un identifiant unique à la section de cours en fonction de ces trois clés primaires pour pouvoir s'en servir comme foreign key plus tard.

Question b

Departement(<u>nomDep</u>)	
Définitions	Tout département à l'école d'ingénierie
Identifiants	nomDep
Identifiant externe	Aucun
Autres contraintes d'intégrité	nomDep not NULL; nomDep unique;

ProgrammeÉtudes(<u>sigleProgramme</u> , nomProgramme, nomDep)	
Définitions	Tout programme dans un département
Identifiants	sigleProgramme
Identifiant externe	nomDep (de Département)
Autres contraintes d'intégrité	sigleProgramme, nomProgramme not NULL; sigleProgramme, nomProgramme unique;

Cours(<u>sigleCours</u> , nomCours, programme)	
Définitions	Tout cours dans un programme
Identifiants	sigleCours
Identifiant externe	programme référence l'attribut sigleProgramme de ProgrammeEtude
Autres contraintes d'intégrité	sigleCours, nomCours not NULL; sigleCours, nomCours unique;

Personne(<u>NAS</u> , prenom, nom, adresse, dateNaissance, sexe, numeroTelephone)	
Définitions	Toute personne quelconque
Identifiants	NAS
Identifiant externe	Aucun
Autres contraintes d'intégrité	NAS not NULL; NAS unique;

Professeur(<u>idProf</u> , NAS, salaire)	
Définitions	Toute personne qui enseigne à l'école d'ingénierie
Identifiants	idProf
Identifiant externe	NAS (de Personne)
Autres contraintes d'intégrité	idProf not NULL; idProf unique; salaire > 0;

Étudiant(<u>idÉtudiant</u> , NAS)	
Définitions	Toute personne qui étudie à l'école d'ingénierie
Identifiants	idÉtudiant
Identifiant externe	NAS (de Personne)
Autres contraintes d'intégrité	idÉtudiant not NULL; idÉtudiant unique;

Épreuve(<u>type</u> , <u>numéroÉpreuve</u> , ponderation, cours, etudiant)	
Définitions	Tout type d'évaluation donnée dans une section de cours
Identifiants	type, numéroÉpreuve

Identifiant externe	Cours référence id de SectionDeCours et etudiant reference idEtudiant d'Étudiant
Autres contraintes d'intégrité	type, numéroÉpreuve not NULL; (type, numéroÉpreuve) unique; ponderation > 0;

SectionDeCours(<u>nomCours</u> , <u>numeroSection</u> , <u>Session</u> , effectifMaximal, id)	
Définitions	Toute classe pour un cours
Identifiants	nomCours, numeroSection, Session
Identifiant externe	nomCours (de Cours)
Autres contraintes d'intégrité	id not NULL; (nomCours, numeroSection, Session), id unique; 0 < effectifMaximal < 200;