

MCD

August 21, 2020

1 Gestion simplifiée des notes d'élèves (R. Hatterer)

Pour mettre `mocodo` à disposition de ce notebook, évaluez la cellule suivante:

```
[1]: %reload_ext mocodo_magic
```

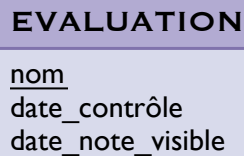
1.1 A. Établissement progressif du Modèle Conceptuel des Données

1.1.1 A.1 Les entités

Je propose de commencer le MCD par l'entité EVALUATION, car avant de reporter les notes de ses élèves dans une application de gestion des notes un professeur commence par donner un nom à l'évaluation, une date (celle où a eu lieu le contrôle), le logiciel peut aussi proposer de saisir la date à partir de laquelle les notes de l'évaluation seront visibles aux élèves.

```
[2]: %%mocodo --colors brewer+4 --shapes copperplate --strengthen_card (1,1)

EVALUATION: nom, date_contrôle, date_note_visible
```



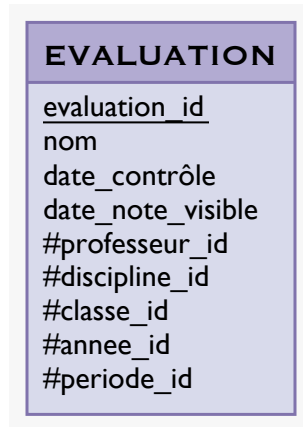
Cependant, ni le nom ni les dates évoquées ne sont pertinents pour identifier l'évaluation. En effet, d'autres professeurs peuvent avoir choisi le même nom, ou le professeur souhaite donner ce nom à une évaluation donnée à une autre classe... On devra utiliser un identifiant unique par exemple (`evaluation_id`) qui sera incrémenté à chaque nouvelle saisie.

De plus, une fois que des évaluations auront été saisies, il faut que les informations qui ont été enregistrées permettent également d'identifier: - le professeur qui a réalisé l'évaluation - la discipline sur laquelle le contrôle portait - la classe à qui était destiné le contrôle - quand il doit être pris en compte (année scolaire, trimestre, semestre...)

En toute logique, les informations devant être portées par l'entité EVALUATION sont donc:

```
[3]: %%mocodo --colors brewer+4 --shapes copperplate --strengthen_card (1,1)

EVALUATION: evaluation_id, nom, date_contrôle, date_note_visible, 
↳#professeur_id, #discipline_id, #classe_id, #annee_id, #periode_id
```



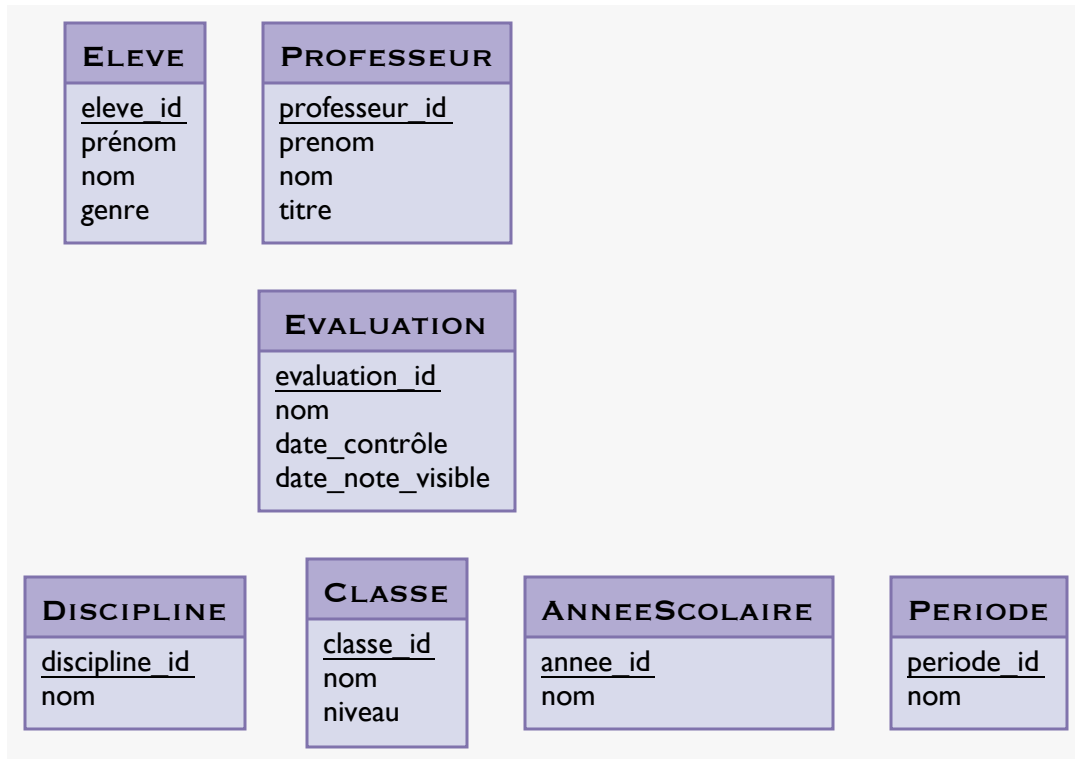
L'analyse précédente a révélé l'existence de nouvelles entités. Chacune portant un certain nombre d'attributs.

```
[4]: %%mocodo --colors brewer+4 --shapes copperplate --relations diagram↳
↳markdown_data_dict--strengthen_card (1,1)

Eleve: eleve_id, prénom, nom, genre
Professeur: professeur_id, prenom, nom, titre
:

Evaluation: evaluation_id, nom, date_contrôle, date_note_visible

Discipline: discipline_id, nom
Classe: classe_id, nom, niveau
AnneeScolaire: annee_id, nom
Periode: periode_id, nom
```



1.1.2 A.2 Les associations

Tâchons d'établir comment l'entité EVALUATION peut interagir avec les autres entités.

Établir des associations entre l'entité EVALUATION et les autres entités doit permettre de retrouver les attributs qui en toute logique doivent aussi être portés par elle.

1. Association ... REALISER par ... Avant de pouvoir noter ses élèves un PROFESSEUR doit REALISER des EVALUATIONS.

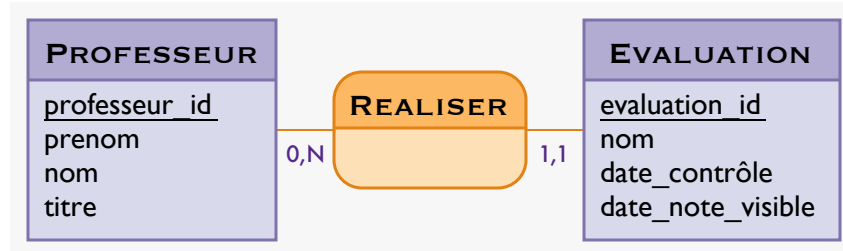
Nous distinguons donc deux **entités** (PROFESSEUR et REALISER) et une **association** (REALISER).

L'**association** REALISER possède les cardinalités suivantes : - Une évaluation est réalisée par **1 et 1** seul professeur - UN professeur réalise **0 ou plusieurs** évaluations

[5] : `%%mocodo --colors brewer+4 --shapes copperplate --strengthen_card (1,1)`

```

Professeur: professeur_id, prenom, nom, titre
Realiser, 11 Evaluation, 0N Professeur
Evaluation: evaluation_id, nom, date_contrôle, date_note_visible
  
```



2. Association ... PORTER sur ... Une EVALUATION va PORTER sur une DISCIPLINE.
Cardinalités: - Une évaluation va porter sur **une et une seule** discipline - Une discipline peut donner lieu à **0 ou plusieurs** évaluations

3. Association ... DESTINER à ... Une EVALUATION est DESTINEE à une CLASSE.
Cardinalités: - Une évaluation est destinée à **une et une seule** classe (rien n'empêche de donner le même sujet une seconde fois, mais le référencement de la seconde évaluation sera différent de la première). - Une classe peut subir **0 ou plusieurs** évaluations.

4. Association SITUER dans le temps Une EVALUATION est à SITUER dans une période scolaire précise (année-scolaire, trimestre ou semestre, examen blanc...)
Cardinalités: - Une évaluation est à situer dans **une et une seule** année-scolaire et dans **une et une seule** période (trimestre ou semestre, examen blanc...) - Une année-scolaire peut donner lieu à **0 ou plusieurs** évaluations. - Il est est de même pour une période.

Une association peut, comme ici, impliquer plus de deux entités.

Situation intermédiaire Le MCD et le MLD correspondant aux **entités** et **associations** précédentes sont donnés par mocodo grâce au code suivant:

```
[6]: %%mocodo --mld --colors brewer+4 --shapes copperplate --relations diagram
↪markdown_data_dict--strengthen_card (1,1)

Discipline: discipline_id, nom
Porter, 11 Evaluation, 0N Discipline
::

:
Professeur: professeur_id, prenom, nom, titre
Realiser, 11 Evaluation, 0N Professeur
Evaluation: evaluation_id, nom, date_contrôle, date_note_visible

:

Classe: classe_id, nom, niveau
Destiner, 11 Evaluation, 0N Classe
:
```

Situer, 0N AnnéeScolaire, 0N Période, 11 Evaluation

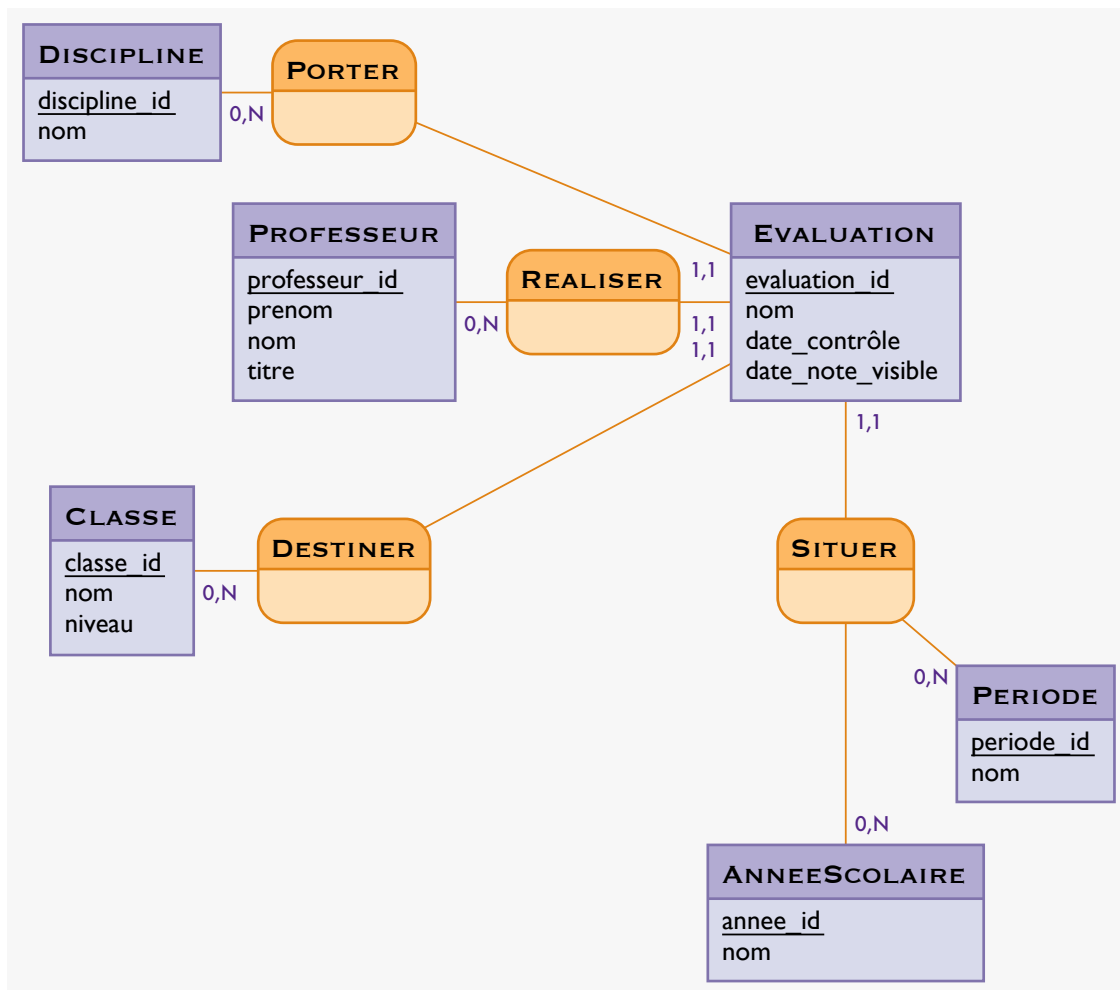
:
:
:
:

Période: periode_id, nom

:::

AnnéeScolaire: annee_id, nom

:



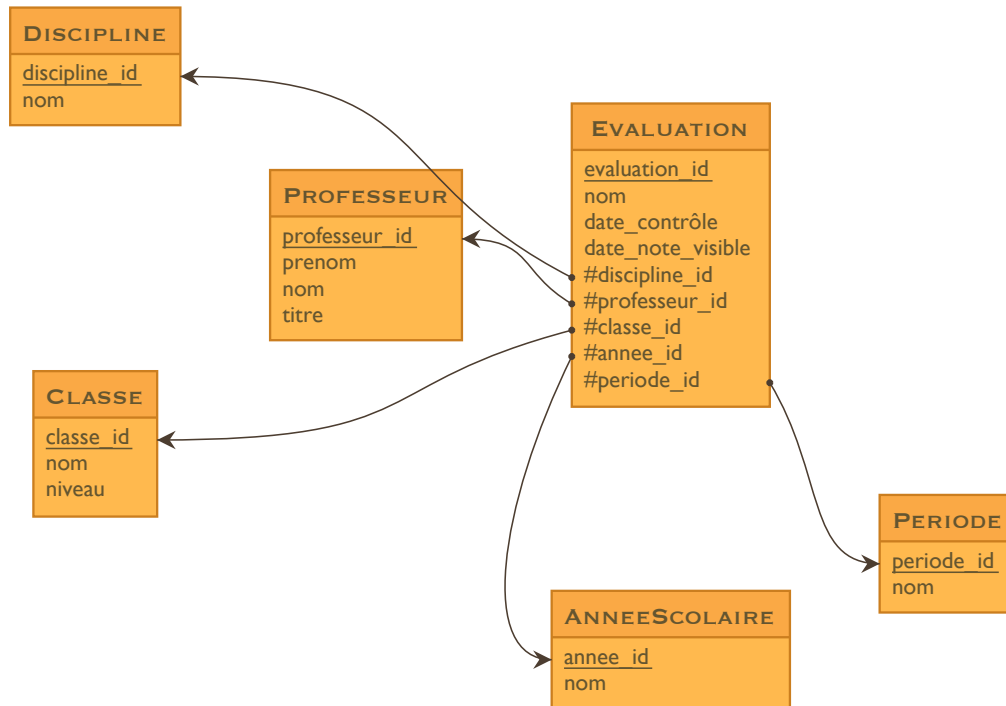
<IPython.core.display.HTML object>

Le MLD ([modèle logique de données](#)) ci-dessus, fait apparaître six **relations**.

Une **relation** possède un nom qui correspond en général à celui de l'**entité** ou de l'**association** qui

lui correspond. Elle possède aussi une clef primaire qui permet d'identifier sans ambiguïté chaque occurrence de cette relation. La clef primaire peut être composée d'un ou plusieurs attributs, il s'agit d'une implantation de la notion d'identifiant des entités et associations qui se répercute au niveau relationnel.

```
[146]: %mocodo --input mocodo_notebook/sandbox.mld --colors desert
```



Dans le MLD établi par Mocodo, la **relation** EVALUATION comporte bien les attributs que nous avons trouvé logique que l'**entité** EVALUATION possédât.

Complétons le MCD en ajoutant l'entité ELEVE et les associations manquantes.

5. Association EVALUER qui ? Une EVALUATION va permettre d'EVALUER des ELEVES et donc de leur attribuer une note.

Cardinalités: - Une évaluation va permettre d'évaluer **un ou plusieurs** élèves. - Un élève va être évalué grâce à **une ou plusieurs** évaluations.

6. Association APPARTENIR Cardinalité: - Un élève appartient à **une et une seule** classe
- Une classe comporte **un ou plusieurs** élèves

7. Association EXISTER Cardinalité: - Durant une année scolaire existent **une ou plusieurs** classes - Une classe existe pendant **une et une seule** année scolaire.

8. Association ENSEIGNER Cardinalité: - Un professeur enseigne **0 ou plusieurs** disciplines à **0 ou plusieurs** classes. - Une discipline est enseignée à **0 ou plusieurs** classes par **0 ou**

plusieurs professeurs (cela couvre le cas où une discipline cesserait d'être enseignée suite à une réforme où à la désaffectation des élèves). - Une classe bénéficie de l'enseignement de **1 ou plusieurs** professeurs enseignant **une ou plusieurs** disciplines.

```
[7]: %%mocodo --mld --colors brewer+4 --shapes copperplate --relations diagram
      ↪markdown_data_dict--strengthen_card (1,1)
```

```
Discipline: discipline_id, nom
Porter, 11 Evaluation, ON Discipline
::

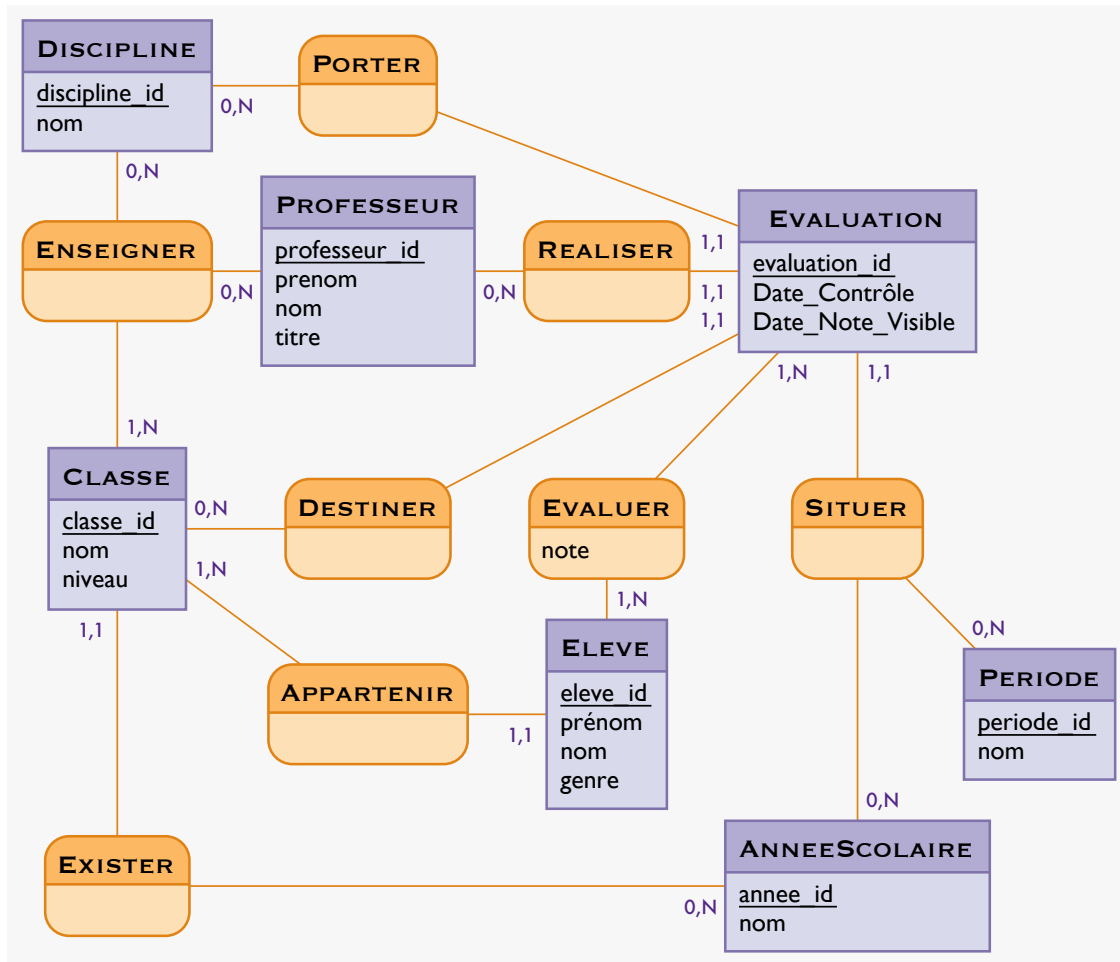
Enseigner, ON Professeur, 1N Classe, ON Discipline
Professeur: professeur_id, prenom, nom, titre
Realiser, 11 Evaluation, ON Professeur
Evaluation: evaluation_id, Date_Contrôle, Date_Note_Visible

:

Classe: classe_id, nom, niveau
Destiner, 11 Evaluation, ON Classe
Evaluer, 1N Evaluation, 1N Eleve: note
Situer, ON AnneeScolaire, ON Periode, 11 Evaluation

:
Appartenir, 1N Classe, 11 Eleve
Eleve: eleve_id, prénom, nom, genre
:
Periode: periode_id, nom

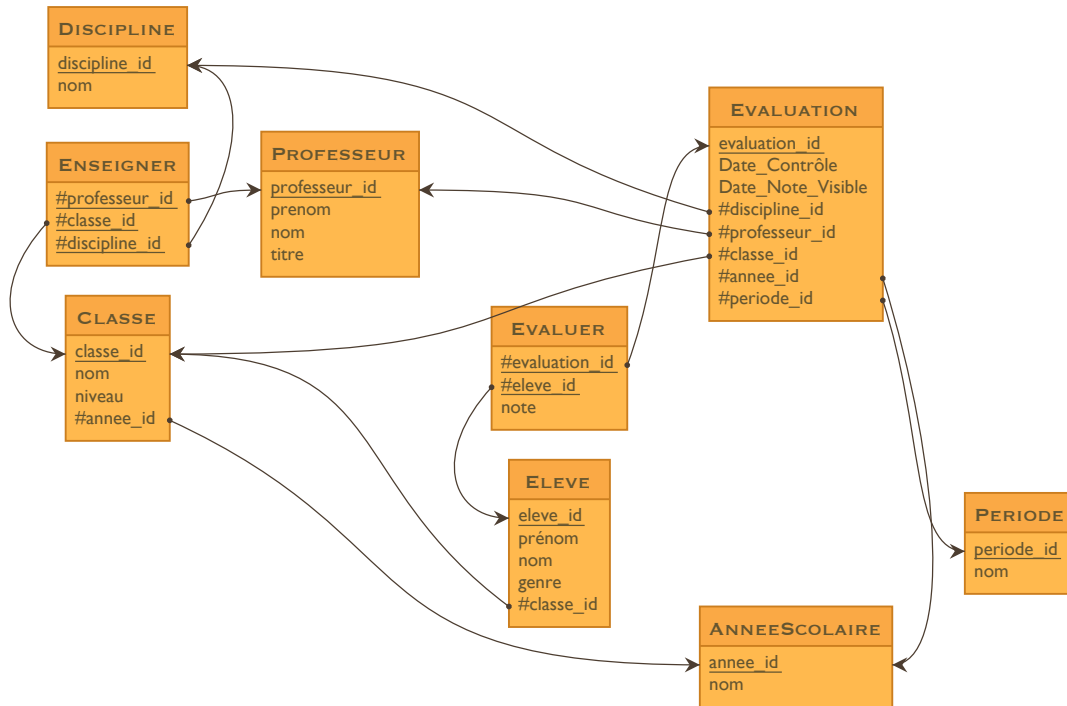
Exister, ON AnneeScolaire, 11 Classe
::
AnneeScolaire: annee_id, nom
:
```



<IPython.core.display.HTML object>

1.2 B. Le modèle logique de données (MLD) obtenu

```
[9]: %mocodo --input mocodo_notebook/sandbox.mld --colors desert --scale 0.8
```

1.3 C. Le code SQL

Précisons les types de données entre crochets après chaque attribut dans le MCD:

```
[155]: %%mocodo --title=mocodo_gestion_notes --relations=mysql --mld
```

```

Discipline: discipline_id, nom
Porter, 11 Evaluation, ON Discipline
::

Enseigner,  ON Professeur, 1N Classe, ON Discipline
Professeur: professeur_id, prenom, nom, titre
Realiser, 11 Evaluation,  ON Professeur
Evaluation: evaluation_id, Date_Contrôle, Date_Note_Visible

:

Classe: classe_id, nom, niveau
Destiner, 11 Evaluation, ON Classe
Evaluer, 1N Evaluation, 1N Eleve: note
Situer,  ON AnneeScolaire,  ON Periode, 11 Evaluation

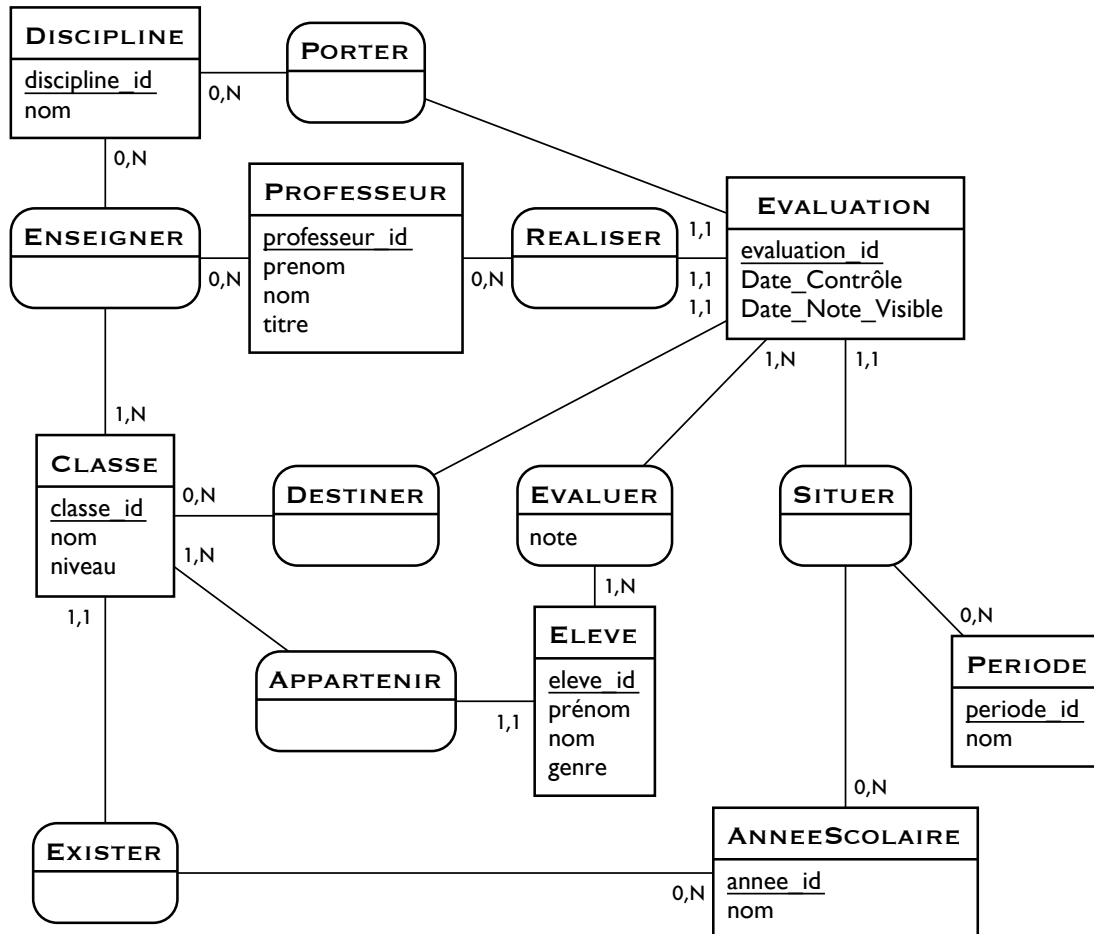
:
Appartenir, 1N Classe, 11 Eleve
  
```

```

Eleve: eleve_id, prénom, nom, genre
:
Periode: periode_id, nom

Exister, ON AnneeScolaire, 11 Classe
::
AnneeScolaire: annee_id, nom
:

```



<IPython.core.display.HTML object>

Si des types ont été oubliés, mocodo remplace les types manquants par VARCHAR(42).

```

[ ]: # %load mocodo_notebook/sandbox_mysql.sql
CREATE DATABASE IF NOT EXISTS `MOCODO_GESTION_NOTES` DEFAULT CHARACTER SET utf8
    ↳COLLATE utf8_general_ci;
USE `MOCODO_GESTION_NOTES`;

```

```

CREATE TABLE `DISCIPLINE` (
  `discipline_id` VARCHAR(42),
  `nom` VARCHAR(42),
  PRIMARY KEY (`discipline_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `ENSEIGNER` (
  `professeur_id` VARCHAR(42),
  `classe_id` VARCHAR(42),
  `discipline_id` VARCHAR(42),
  PRIMARY KEY (`professeur_id`, `classe_id`, `discipline_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `PROFESSEUR` (
  `professeur_id` VARCHAR(42),
  `prenom` VARCHAR(42),
  `nom` VARCHAR(42),
  `titre` VARCHAR(42),
  PRIMARY KEY (`professeur_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `EVALUATION` (
  `evaluation_id` VARCHAR(42),
  `date_contrôle` VARCHAR(42),
  `date_note_visible` VARCHAR(42),
  `discipline_id` VARCHAR(42),
  `professeur_id` VARCHAR(42),
  `classe_id` VARCHAR(42),
  `annee_id` VARCHAR(42),
  `periode_id` VARCHAR(42),
  PRIMARY KEY (`evaluation_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `CLASSE` (
  `classe_id` VARCHAR(42),
  `nom` VARCHAR(42),
  `niveau` VARCHAR(42),
  `annee_id` VARCHAR(42),
  PRIMARY KEY (`classe_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

```

```

CREATE TABLE `EVALUER` (
  `evaluation_id` VARCHAR(42),
  `eleve_id` VARCHAR(42),
  `note` VARCHAR(42),
  PRIMARY KEY (`evaluation_id`, `eleve_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `ELEVE` (
  `eleve_id` VARCHAR(42),
  `prenom` VARCHAR(42),
  `nom` VARCHAR(42),
  `genre` VARCHAR(42),
  `classe_id` VARCHAR(42),
  PRIMARY KEY (`eleve_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `PERIODE` (
  `periode_id` VARCHAR(42),
  `nom` VARCHAR(42),
  PRIMARY KEY (`periode_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

CREATE TABLE `ANNEESCOLAIRE` (
  `annee_id` VARCHAR(42),
  `nom` VARCHAR(42),
  PRIMARY KEY (`annee_id`)
) ENGINE=InnoDB DEFAULT CHARSET=utf8;

ALTER TABLE `ENSEIGNER` ADD FOREIGN KEY (`discipline_id`) REFERENCES `DISCIPLINE` (`discipline_id`);
ALTER TABLE `ENSEIGNER` ADD FOREIGN KEY (`classe_id`) REFERENCES `CLASSE` (`classe_id`);
ALTER TABLE `ENSEIGNER` ADD FOREIGN KEY (`professeur_id`) REFERENCES `PROFESSEUR` (`professeur_id`);
ALTER TABLE `EVALUATION` ADD FOREIGN KEY (`periode_id`) REFERENCES `PERIODE` (`periode_id`);
ALTER TABLE `EVALUATION` ADD FOREIGN KEY (`annee_id`) REFERENCES `ANNEESCOLAIRE` (`annee_id`);
ALTER TABLE `EVALUATION` ADD FOREIGN KEY (`classe_id`) REFERENCES `CLASSE` (`classe_id`);
ALTER TABLE `EVALUATION` ADD FOREIGN KEY (`professeur_id`) REFERENCES `PROFESSEUR` (`professeur_id`);

```

```

ALTER TABLE `EVALUATION` ADD FOREIGN KEY (`discipline_id`) REFERENCES `DISCIPLINE` (`discipline_id`);
ALTER TABLE `CLASSE` ADD FOREIGN KEY (`annee_id`) REFERENCES `ANNEESCOLAIRE` (`annee_id`);
ALTER TABLE `EVALUER` ADD FOREIGN KEY (`eleve_id`) REFERENCES `ELEVE` (`eleve_id`);
ALTER TABLE `EVALUER` ADD FOREIGN KEY (`evaluation_id`) REFERENCES `EVALUATION` (`evaluation_id`);
ALTER TABLE `ELEVE` ADD FOREIGN KEY (`classe_id`) REFERENCES `CLASSE` (`classe_id`);

```

1.4 Rôle de l'administrateur

Ne relève pas du MCD, mais mocodo est un outil pratique pour indiquer le rôle de l'administrateur de cette base de donnée.

```

[152]: %/mocodo --mld --colors brewer+4 --shapes copperplate --relations diagram
       ↪ markdown_data_dict--strengthen_card (1,1)

```

```

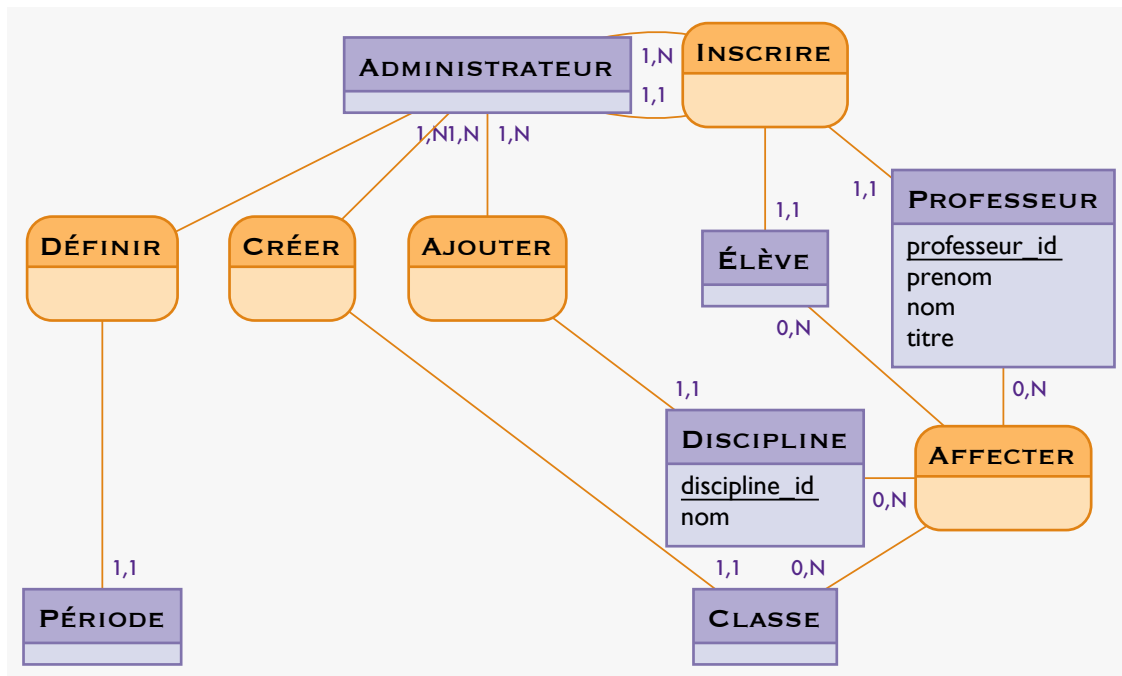
:
Administrateur:
Inscrire, 1N Administrateur, 11 Administrateur, 11 Professeur, 11 Élève

Définir, 11 Période, 1N Administrateur
Créer, 11 Classe, 1N Administrateur
Ajouter, 11 Discipline, 1N Administrateur
Élève:
Professeur : professeur_id, prenom, nom, titre

:::
Discipline : discipline_id, nom
Affecter, ON Professeur, ON Discipline, ON Classe, ON Élève

Période:
::
Classe:

```



<IPython.core.display.HTML object>

[]: