

# Base De Donnée

IFT 2935

Thème : Stage

Lalla Diakite : 20256054

Twendo-Raab, Fahima Carmen Dabo : 20266362

Wendkounie Reine Roseine Guinguere : 20231523

Mounir Ammam : 20120442

Présenté à : Michel Boyer

Faculté des Arts et des Sciences

Université De Montréal

27 - 04 - 2025

# 1- Modélisation

## Stages

Une école organise et gère les stages de ses étudiants.

Chaque année, l'école recherche des stages pour les étudiants. Pour cela, l'école dispose d'une liste d'entreprises ayant déjà proposé des stages et d'autres susceptibles d'en proposer, et leur écrit pour les inciter à renouveler l'expérience. Au vu des réponses, les étudiants prennent contact avec les entreprises qui les intéressent. Quand les conventions de stage sont signées, chaque stagiaire se voit désigner un enseignant pour l'encadrer. Ce dernier doit veiller au bon déroulement du stage et aller rendre visite à l'étudiant dans son entreprise d'accueil. Une entreprise peut offrir plusieurs stages. Un enseignant peut encadrer plusieurs étudiants. Un stage est effectué dans une seule entreprise et est encadré par un seul enseignant. Pour ses archives, l'école conserve toutes les informations concernant les stages effectués par ses étudiants. Les stages ont une date de début et une date de fin.

On pourra inclure dans chaque entité tous les éléments réalistes possibles. Justifier les choix.

Pour ce projet voici la liste de nos Entités

**Etudiants:** l'école gère les stages de ses étudiants en doit donc identifier chaque étudiant de manière unique et suivre leur Parcours

- Matricule\_et
- Nom
- Prenom
- Email\_et
- DDN
- Programme
- Nbre de Credits (il faut avoir Validé un certains credit avant d'être éligible à un stage)

**Entreprise:** Chaque entreprises peut proposer ou non des stages. L'adresse et le contact sont nécessaire pour permettre la communication

- ZD-Entreprise
- Nom
- Adresse
- Courriel
- Num\_tel

**Enseignants:** chaque enseignant encadre un ou plusieurs étudiants on doit donc garder une trace de qui encadre qui.

- Matricule\_ens
- Nom
- Prenom
- Email\_ens
- Faculté (ex: science, littérature)

**Stage:** le stage est une entité centrale car c'est grâce à ses offres que les étudiants peuvent postuler

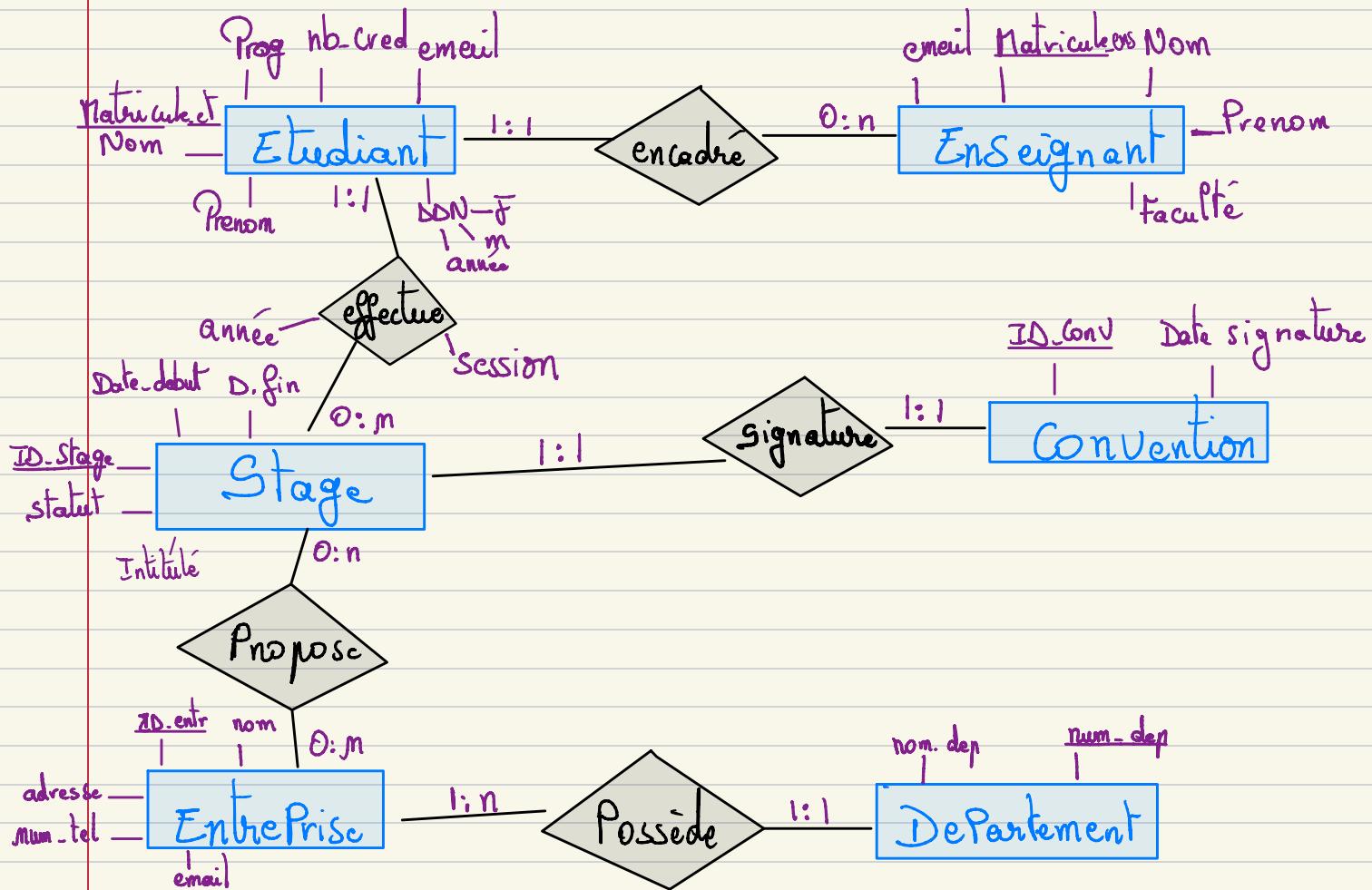
- Date Début
- Date Fin
- ID Stage
- Statut Stage (ouvert, Fermé, en cours, Annulé)

**Convention:** le contrat officiel entre l'étudiant et l'entreprise nécessaire pour l'archivage et la légalité

- ID\_Convention
- Date Signature

**Département:** Etant donné qu'il y'a plusieurs secteurs d'activités dans une entreprise (Finance, Informatique...)

- Num. Département
- Nom\_Département



## Explication de notre Modélisation: Gestion des Stages

• L'étudiant est un acteur central du modèle, car ce sont les stages des étudiants que l'école organise et suit: chaque étudiant est identifié par un matricule unique, on conserve des informations personnelles comme: nom, prénom, date de naissance (DDN), email. Les informations académiques comme le programme et le nombre de crédit sont aussi nécessaires pour déterminer son éligibilité aux stages. Nous avons l'association "Encadre" entre étudiant et enseignant de cardinalité 1:1 pour étudiant car un étudiant à un seul encadrant, et 0:n pour enseignant car un enseignant peut encadrer plusieurs étudiants.

• L'enseignant encadrent les étudiants pendant leurs stages. chaque enseignant est identifié par son matricule et on conserve son nom, prénom, email et la faculté à laquelle il appartient (ex: Faculté des arts...). Un enseignant peut encadrer plusieurs étudiants, mais chaque étudiant est encadré par un seul enseignant.

• Les entreprises sont les lieux d'accueil pour les stages. chaque entreprise a un identifiant (ID\_entreprise), un nom, une adresse, un courriel et un N°tel. Une entreprise peut me proposer plusieurs stages dans différents départements ou périodes. les entreprises possèdent un ou plusieurs départements (1:n)

• Le stage est aussi un élément clé de cette base. chaque stage est défini par une date de début de fin, un identifiant (ID\_stage) et un statut (ex: ouvert, Fermé, Annulé...). Un stage peut recevoir un ou plusieurs étudiants, proposer par une entreprise et encadré par un enseignant. L'association signature est un lien entre stage et convention de cardinalité (1:1) les deux car chaque stage donne lieu à une seule convention (contrat).

• La convention représente le contrat officiel entre l'étudiant, l'entreprise et l'école. Elle contient un identifiant (ID\_Convention). Elle est signée pour chaque stage garantissant la légalité et l'archivage.

• Les départements sont les différents secteurs d'activité des entreprises (ex: Finance, info..) chaque département est identifié par un nom et un numéro

Avec ce modèle on peut suivre le parcours de chaque étudiant, les stages réalisés, les entreprises partenaires, les enseignants encadrants ainsi que les conventions signées pour chaque stage.

## 2- Transformation

Stage: ID-stage, date-debut, date-fin, statut, #ID-conv, intitulé-stage

Convention: ID-conv, D-signature

Enseignant: Matricule-ens, Nom, Prenom, email-ens, Faculté

Etudiant: Matricule-et, Nom, Prenom, date-naissance, email, Programme, Nb.credit, #ID-stage, #Matricule-ens, année-stage, session-stage

Département: Num-departement, Nom-departement, #ID-entreprise

Entreprise: ID-entreprise, Nom-entr, N°rue, Ville, Province, CP, email, N°tel

Propose: #ID-entreprise, #ID-stage

### Explication

Chaque entité forte du modèle E/A devient une relation avec sa clé primaire et ses attributs

La clé primaire de stage est id-stage et la comme clé étrangère #ID-conv venant de l'association avec convention à cause du lien 1:1 entre les deux l'une des deux entités doit donc avoir la clé primaire de l'autre.

- La clé primaire de convention est id-conv et il garde ses autres attributs
- L'enseignant aura comme clé primaire Matricule-ens et ses autres attributs
- L'étudiant aura ses autres attributs et sa clé primaire Matricule-et, on y rajoute la clé étrangère #ID-stage étant de cardinalité 1:1 en lien avec stage, #Matricule-ens étant également de cardinalité 1:1 en lien avec enseignant, année-stage et session-stage viennent de l'association effectue qui a ses attributs.
- Département garde ses attributs et comme clé primaire Num-departement. La clé étrangère #ID-entreprise vient de son lien 1:1 avec entreprise
- L'entreprise gardera également ses attributs et comme clé primaire id-entreprise.

### 3- Normalisation

#### • Dépendance Fonctionnelle

Stage:  $\{ \text{id\_stage}, \text{date\_debut}, \text{date\_fin}, \text{statut\_stage}, \# \text{id\_conv}, \text{intitulé\_stage} \}$

Dépendance fonctionnelle:  $\{ \text{id\_stage} \rightarrow \text{date\_debut}, \text{date\_fin}, \text{statut\_stage}, \# \text{id\_conv}, \text{intitulé\_stage}$   
 $\text{id\_conv} \rightarrow \text{intitulé\_stage}, \text{date\_debut}, \text{date\_fin} \}$

Convention:  $\text{id\_conv}, \text{D\_signature}$

Dépendance fonctionnelle:  $\{ \text{id\_conv} \rightarrow \text{D\_signature} \}$

Enseignant:  $\text{Matricule\_ens}, \text{Nom}, \text{Prenom}, \text{email\_ens}, \text{Faculté}$

Dépendance fonctionnelle:  $\{ \text{Matricule\_ens} \rightarrow \text{Nom}, \text{Prenom}, \text{email\_ens}, \text{Faculté}$   
 $\text{email\_ens} \rightarrow \text{Nom}, \text{Prenom}, \text{Faculté} \}$

Etudiant:  $\text{Matricule\_et}, \text{Nom}, \text{Prenoms}, \text{f\_naiss}, \text{mois\_naiss}, \text{ann\_naiss}, \text{email}, \text{Programme}, \text{Nb\_credit}$   
 $\# \text{id\_stage}, \# \text{Matricule\_ens}, \text{année\_stage}, \text{session\_stage}$

Dépendance fonctionnelle:  $\{ \text{Matricule\_et} \rightarrow \text{Nom}, \text{Prenoms}, \text{date\_naissance}, \text{email}$   
 $\text{Programme}, \text{Nb\_credit}, \# \text{id\_stage}, \# \text{Matricule\_ens}$   
 $\text{année\_stage}, \text{session\_stage} \}$

$\text{email\_et} \rightarrow \text{Nom}, \text{Prenom} \}$

Département:  $\text{Num\_département}, \text{Nom\_département}, \# \text{id\_entreprise}$

Dépendance fonctionnelle:  $\{ \text{Num\_département} \rightarrow \text{Nom\_département}, \text{id\_entreprise} \}$

Entreprise:  $\text{id\_entreprise}, \text{Nom\_entr}, \text{N°\_rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email}, \text{N°\_tel}$

Dépendance fonctionnelle:  $\{ \text{id\_entreprise} \rightarrow \text{Nom\_entr}, \text{N°\_rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email}, \text{N°\_tel}$   
 $\text{Nom\_entr} \rightarrow \text{N°\_rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email}, \text{N°\_tel}$   
 $\text{CP} \rightarrow \text{Ville}, \text{Province} \}$

Si on prend pour acquis que c'est le QG général de l'entreprise qui poste l'offre de stage donc son nom va renvoyer à un unique ID quitte à ce que le QG assigne chaque stagiaire à ses différentes succursales

Propose: n'a pas de dépendances fonctionnelles car aucun attribut ne permet de définir un autre attribut

# Projet (Partie normalisation)

## Stage

$\mathcal{F} = \{ idStage \rightarrow^{\textcircled{2}} intituleStage, dateDebut, dateFin, statutStage, idConu;$   
 $intituleStage, idStage \rightarrow^{\textcircled{3}} statutStage;$   
 $idConu \rightarrow^{\textcircled{3}} idStage, intituleStage, dateDebut, dateFin, statutStage\}$

- ② La DF  $intitule, idStage \rightarrow statutStage$  est non élémentaire car on peut obtenir le statut du stage grâce à son identifiant. Cette DF peut donc être supprimée.

$$\{ intituleStage \}^+ = \{ intituleStage \}$$

$$\{ idStage \}^+ = \{ idStage, intituleStage, dateDebut, dateFin, statutStage, idConu \}$$

$statutStage \in \{ idStage \}^+$  donc la DF est non élémentaire.

- ③ D'après le modèle E/A ; un stage est lié à une seule convention et une convention est liée à un seul stage. Par conséquent l'identifiant d'une convention permet de déduire une instance unique de la relation Stage.

On a donc :

$$\mathcal{F} = \{ idStage \rightarrow intituleStage, dateDebut, dateFin, statutStage, idConu;$$
  
 $idConu \rightarrow idStage, intituleStage, dateDebut, dateFin, statutStage \}$

3) Trouvons les clés de la relation

$\{ \text{idStage} \}^+ = \{ \text{idStage}, \text{intituléStage}, \text{dateDebut}, \text{dateFin}, \text{statutStage}, \text{idConvo} \} = \text{Stage}$

$\{ \text{idConvo} \}^+ = \{ \text{idConvo}, \text{idStage}, \text{intituléStage}, \text{dateDebut}, \text{dateFin}, \text{statutStage} \} = \text{Stage}.$

$\Rightarrow$  idStage et idConvo sont toutes les deux des clés de la relation Stage. On va choisir idStage comme clé primaire et idConvo comme clé candidate.  
Cela nous amène à la couverture minimale suivante :

$\mathcal{G} = \{ \text{idStage} \rightarrow \text{intituléStage}, \text{dateDeb}, \text{dateFin}, \text{statutStage}, \text{idConvo} \}$

4) Normaliser la table en 3NF :

Algô de Bernstein :

1. Crée une relation pour chaque X distinct à grande

Couverture minimale	"factornisation"	Relation
$\text{idStage} \rightarrow \text{intituléStage}, \text{dateDeb}, \text{dateFin}, \text{statutStage}, \text{idConvo}$	$\text{idStage} \rightarrow \text{intituléStage}, \text{dateDebut}, \text{dateFin}, \text{statutStage}, \text{idConvo}$	$R_1 (\text{idStage}, \text{intituléStage}, \text{dateDeb}, \text{dateFin}, \text{statutStage}, \text{idConvo})$

2. Crée une relation avec les clés manquantes.  
idConvo est déjà dans la relation  $R_1$ , donc pas besoin d'ajouter une autre relation.

Conclusion :  $R_1 (\underline{\text{idStage}}, \text{intituléStage}, \text{dateDeb}, \text{dateFin}, \text{statutStage}, \# \text{idConvo})$

## Enseignant :

$$\mathcal{F} = \{ \text{matricule\_ens} \rightarrow \text{nom}, \text{prenom}, \text{email\_ens}, \text{faculté} \\ \text{email\_ens} \rightarrow \text{nom}, \text{prenom} \}$$

### 1 - Couverture minimale

a) Décomposer les termes de droite :

$$\mathcal{F} = \{ \text{matricule\_ens} \rightarrow \text{nom}; \text{matricule\_ens} \rightarrow \text{prenom}; \\ \text{matricule\_ens} \rightarrow \text{email\_ens}; \text{matricule\_ens} \rightarrow \text{faculté} \\ \text{email\_ens} \rightarrow \text{nom}, \text{email\_ens} \rightarrow \text{prenom} \}$$

b) Supprimer les dépendances non élémentaire.

Dans toutes les dépendances fonctionnelles, aucune n'est composée dans la partie de gauche. Donc il n'y a pas de dépendance non-élémentaire à supprimer.

c) Supprimer les dépendances redondantes.

- $\text{email\_ens} \rightarrow \text{nom}$  : on a déjà la dépendance  $\text{matricule} \rightarrow \text{nom}$  donc on peut supprimer cette dépendance.

- $\text{email\_ens} \rightarrow \text{prenom}$  :  $\text{matricule\_ens} \rightarrow \text{prenom}$  donc on peut aussi supprimer cette dépendance.

$$\mathcal{G} = \{ \text{matricule\_ens} \rightarrow \text{nom}; \text{matricule\_ens} \rightarrow \text{prenom}; \\ \text{matricule\_ens} \rightarrow \text{email\_ens}; \text{matricule\_ens} \rightarrow \text{faculté} \}$$

### 2) Bernstein

Couverture minimal

"Factorisation"

Relation

$\text{matricule\_ens} \rightarrow \text{nom}$   
 $\text{matricule\_ens} \rightarrow \text{prenom}$   
 $\text{matricule\_ens} \rightarrow \text{email\_ens}$   
 $\text{matricule\_ens} \rightarrow \text{faculté}$

$\text{matricule\_ens} \rightarrow \text{nom, prenom, email\_ens, faculté}$

$R_1(\text{matricule\_ens, nom, prenom, email\_ens, faculté})$

Conclusion : Enseignant (matricule\_ens, nom, prenom, email\_ens, faculté)

# Etudiant.

$F = \{ \text{matricule\_et} \rightarrow \text{nom, prenom, date\_naissance, email\_et, programme, nb\_credit, id\_stage, matricule\_ens, annee\_stage, session\_stage};$   
 $\text{email\_et} \rightarrow \text{nom, prenom} \}$

## 1) Couverture minimale

### a) Décomposer les termes de droite

$F' = \{ \text{matricule\_et} \rightarrow \text{nom}; \text{matricule\_et} \rightarrow \text{prenom};$   
 $\text{matricule\_et} \rightarrow \text{date\_naissance}; \text{matricule\_et} \rightarrow \text{email\_et}$   
 $\text{matricule\_et} \rightarrow \text{programme}; \text{matricule\_et} \rightarrow \text{nb\_credit}$   
 $\text{matricule\_et} \rightarrow \text{id\_stage}; \text{matricule\_et} \rightarrow \text{matricule\_ens}$   
 $\text{matricule\_et} \rightarrow \text{annee\_stage};$   
 $\text{matricule\_et} \rightarrow \text{session\_stage}; \text{email\_et} \rightarrow \text{nom};$   
 $\text{email\_et} \rightarrow \text{prenom} \}$

### b) Supprimer les dépendances non élémentaire.

Dans toutes les dépendances fonctionnelles, aucun n'est composé dans la partie de gauche. Donc il n'y a pas de dépendance non-élémentaire à supprimer.

### c) Supprimer les redondances

Les dépendances  $\text{email\_et} \rightarrow \text{nom}$  et  $\text{email\_et} \rightarrow \text{prenom}$  sont redondantes car à partir du matricule\\_et on peut avoir le nom et le prénom de l'étudiant.  
On peut donc les supprimer.

$G = \{ \text{matricule\_et} \rightarrow \text{nom}; \text{matricule\_et} \rightarrow \text{prenom};$   
 $\text{matricule\_et} \rightarrow \text{date\_naissance}; \text{matricule\_et} \rightarrow \text{email\_et}$   
 $\text{matricule\_et} \rightarrow \text{programme}; \text{matricule\_et} \rightarrow \text{nb\_credit}$   
 $\text{matricule\_et} \rightarrow \text{id\_stage}; \text{matricule\_et} \rightarrow \text{matricule\_ens}$   
 $\text{matricule\_et} \rightarrow \text{annee\_stage};$   
 $\text{matricule\_et} \rightarrow \text{session\_stage} \}$

$\{ \text{matricule\_et} \}^+ = \{ \text{matricule\_et, nom, prénom, email\_et, id\_stage, programme, mb\_credit, matricule\_ens, année\_stage, session\_stage} \}$

$\{ \text{matricule\_et} \}^+$  = la relation initiale. matricule\\_et est donc une clé de la relation.

## 2) Bernstein

couverture minimal	"Factorisation"	Relation
$\text{matricule\_et} \rightarrow \text{nom};$ $\text{matricule\_et} \rightarrow \text{date\_naissance};$ $\text{matricule\_et} \rightarrow \text{programme};$ $\text{matricule\_et} \rightarrow \text{id\_stage};$ $\text{matricule\_et} \rightarrow \text{année\_stage};$ $\text{matricule\_et} \rightarrow \text{session\_stage}$ $\text{matricule\_et} \rightarrow \text{prénom};$ $\text{matricule\_et} \rightarrow \text{email\_et}$ $\text{matricule\_et} \rightarrow \text{mb\_credit}$ $\text{matricule\_et} \rightarrow \text{matricule\_ens}$	$\text{matricule\_et} \rightarrow$ $\text{nom, prénom, date\_naissance, email\_et, programme, mb\_credit, id\_stage, matricule\_ens}$	$R, (\text{matricule\_et, nom, prénom, date\_naissance, email\_et, programme, mb\_credit, id\_stage, matricule\_ens})$

$\Rightarrow$  la clé se retrouve dans la R, donc pas besoin d'ajouter une autre relation.

**Conclusion :** Etudiant (matricule\\_et, nom, prénom, date\\_naissance, email\\_et, programme, mb\\_credit, id\\_stage, matricule\\_ens )  
(année\\_stage, session\\_stage)

## De partement

$F = \{ \text{num\_departement} \rightarrow \text{nom\_departement}$   
 $\quad \quad \quad \text{num\_departement} \rightarrow \text{id\_entreprise} \}$

$\{ \text{num\_departement} \}^+ = \{ \text{num\_departement, nom\_departement, id\_entreprise} \}$

$\Rightarrow$  num\\_departement engendre toute la relation donc est une clé. Ici la relation est 3NF car il n'y a pas d'attribut non-clé qui dépendent d'un attribut non-clé.

**Conclusion :** Département (num-departement  
nom département, id-entreprise).

**Convention :**

$$f = \{ \text{id\_conv} \rightarrow D\text{-signature} \}$$

On a une seule dépendance fonctionnelle  
dans la relation donc la convention est déjà  
normalisée.

**Conclusion :** Convention (ID-conv, D-signature)

## Entreprise:

### 1- Signification des attributs de la relation

↳ ID-entreprise : identifiant unique de l'entreprise

↳ Nom-ent : nom de l'entreprise

↳ adresse : N°, rue, Ville, Province, CP, email de l'entreprise

↳ email-ent : adresse courriel de l'entreprise

↳ N°tel : numero de telephone de l'entreprise

### 2- Dépendance fonctionnelles que l'on peut tirer de la relation entreprise

$$\begin{aligned} \text{G}^1 = \{ & \text{id\_entreprise} \xrightarrow{\textcircled{1}} \text{Nom\_entr}, \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email\_ent}, \text{N}^{\circ}\text{tel} \\ & \text{Nom\_entr} \xrightarrow{\textcircled{2}} \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email}, \text{N}^{\circ}\text{tel}, \text{id\_entreprise} \\ & \text{CP} \xrightarrow{\textcircled{3}} \text{Ville}, \text{Province} \} \end{aligned}$$

### 3- les clés de la Relation

$$\begin{aligned} \{ \text{id\_entreprise} \}^+ &= \{ \text{Nom\_ent}, \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email\_ent}, \text{N}^{\circ}\text{tel}, \text{id\_entreprise} \} \\ \{ \text{Nom\_ent} \}^+ &= \{ \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email}, \text{N}^{\circ}\text{tel}, \text{id\_entreprise}, \text{Nom\_ent} \} \\ \{ \text{CP} \}^+ &= \{ \text{CP}, \text{Ville}, \text{Province} \} \\ \bullet \text{id\_entreprise} &(\text{clé Primaire}) \\ \bullet \text{Nom\_ent} &(\text{clé candidate}) \end{aligned}$$

$$\begin{aligned} \text{G}^2 = \{ & \text{id\_entreprise} \xrightarrow{\textcircled{1}} \text{Nom\_entr}, \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email\_ent}, \text{N}^{\circ}\text{tel} \\ & \text{Nom\_entr} \xrightarrow{\textcircled{2}} \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email}, \text{N}^{\circ}\text{tel}, \text{id\_entreprise} \\ & \} \end{aligned}$$

### 4) Normaliser la table en 3NF:

Algo de Bernstein

1- Creer une relation pour chaque x distinct à gauche

$$R_1 = (\text{Nom\_ent}, \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email\_ent}, \text{N}^{\circ}\text{tel}, \text{id\_entreprise})$$

$$R_2 = (\text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email}, \text{N}^{\circ}\text{tel}, \text{id\_entreprise}, \underline{\text{Nom\_ent}})$$

### 5- Identifier les clés dans chaque relation

Dans R<sub>1</sub>, la clé est id-entreprise

Dans R<sub>2</sub>, la clé est Nom-entreprise

Conclusion:

Pour voyons que R<sub>1</sub> et R<sub>2</sub> ont les mêmes attributs alors R<sub>2</sub> est redondante et par transitivité R<sub>1</sub> le donne donc notre clé sera:

$$R_1 = (\text{Nom\_ent}, \text{N}^{\circ}, \text{rue}, \text{Ville}, \text{Province}, \text{CP}, \text{email\_ent}, \text{N}^{\circ}\text{tel}, \underline{\text{id\_entreprise}})$$

## Relations finales :

Etudiant ( matricule\_et , nom , prenom , date\_naissance , email\_et , programme , mbs\_credit , # id\_stage , # matricule\_ens )

Département ( num\_departement , nom\_departement , # id\_entreprise )

Propose ( # id\_entreprise , # id\_stage )

Enseignant ( matricule\_ens , nom , prenom , email\_ens , faculte )

Stage ( id\_stage , intitule\_stage , date\_debut , date\_fin , statut\_stage , # id\_conv )

Convention ( id\_conv , d\_signature )

Entreprise : ( Nom\_ent , N° , rue , Ville , Province , CP , email\_ent , N°tel , # ID\_entreprise )

Propose: # ID\_entreprise , # ID\_stage

## 5- Question / Réponse

• **Question:** Listez les stages qui n'ont pas encore été comblé (statut "Ouvert")

- Algèbre relationnelle :

Relation concernée

Stage: (Id\_stage, intitulé\_stage, dateDebut, dateFin, statut\_stage, id\_conv)

Convention (id\_conv, d\_signature)

$R_1 = \sigma_{\text{statut\_stage} = \text{"Ouvert}}(\text{Stage})$

$R_2 = \pi_{\text{id\_stage}, \text{intitulé\_stage}, \text{id\_conv}}(R_1)$

$R_3 = \pi_{\text{id\_conv}}(\text{Convention})$

$R_4 = R_2 \bowtie R_3$

Result =  $\pi_{\text{id\_stage}, \text{intitulé\_stage}}(R_4)$

- SQL

Select s.id\_stage, s.intitulé\_stage

From stage s

Left Join convention c ON s.IdConv = c.Id\_Conv

Where statut\_stage = "Ouvert"

Optimisation: ici l'utilisation de Left Join au lieu d'un join normal

l'utilisation de Join à la fin après une sélection

Question 2: Trouver les étudiants en stage dans un programme en lien avec ex "Informatique"

Relation concernée

Etudiant : Attributs Pertinents: Nom, Prenom, Programme, matricule\_et

- Algèbre relationnelle :

$R_1 = \sigma_{\text{programme} = \text{"Informatique}}(\text{Etudiant})$

Filtrer pour réduire le nombre de ligne

Result =  $\pi_{\text{nom}, \text{prenom}, \text{programme}, \text{matricule\_et}}(R_1)$

Optimisation: on applique la sélection d'abord afin de limiter les données manipulées

SQL:

Select nom, prenom, programme, matricule\_et

From Etudiant

Where programme ILike '%informatique%';

Question 3: Obtenir les étudiants encadrés par un enseignant donné ex "Diakité"

Relation concernée

Etudiant : Attributs pertinents : matricule\_et, nom, Prenom, matricule\_ens

Enseignant: Attributs pertinents : matricule\_ens, nom, Prenom, email, faculté

- Algèbre relationnelle :

$R_1 = \sigma_{\text{nom} = \text{"Diakité}}(\text{Enseignant})$

Selection pour réduire le nombre de ligne à joindre

$R_2 \leftarrow \text{Etudiant} \setminus \{\text{Etudiant}. \text{matricule\_ens} = R_1. \text{matricule\_ens}\} R_1$

$\text{Result} \leftarrow \text{JT}_{\text{mom}}, \text{prenom}, \text{matricule\_et} \quad (R_2)$

Optimisation: Sélection pour réduire le nombre de ligne à joindre

SQL:

Select e.matricule, e.nom, e.prenom

From Etudiant e

Join Enseignant ens ON e.matricule\_ens = ens.matricule\_ens

Where ens.nom = "Diakite"

Question 4: Quels enseignants encadrent au moins 2 étudiants inscrit dans des stages différents?

Relation concernée

Etudiant: Attribut Pertinent: (matricule\_et, #id\_stage, matricule\_ens)

Enseignant: Attribut Pertinent: (matricule\_ens, nom, Prenom)

Stage: id\_stage, intitule\_stage

-Algèbre relationnelle:

$R_1 \leftarrow \text{JT}_{\text{matricule\_et}, \text{id\_stage}, \text{matricule\_ens}} (\text{Etudiant})$

Grouper les étudiants par enseignant + stage

$R_2 \leftarrow V_{\text{matricule\_ens}, \text{id\_stage}}; \text{Count}(\text{matricule\_et}) \rightarrow \text{nb\_et} (R_1)$

Compter le nombre de stages différents par enseignant

$R_3 \leftarrow V_{\text{matricule\_ens}}, \text{Count}(\text{id\_stage}) \rightarrow \text{nb\_stages} (R_1)$

Garder ceux qui encadrent au moins 2 stages différents

$R_4 \leftarrow \text{T}_{\text{nb\_stages} \geq 2} (R_3)$

Rejoindre avec la Table Enseignant pour obtenir les noms

$R_5 \leftarrow R_4 \bowtie \text{Enseignant}$

$\text{Result} \leftarrow \text{JT}_{\text{matricule\_ens}, \text{nom}, \text{prenom}} (R_5)$

SQL:

Select ens.matricule\_ens, ens.nom, ens.prenom

From Etudiant e

Join Etudiant ens ON e.matricule\_ens = ens.matricule\_ens

Group By ens.matricule\_ens, ens.nom, ens.prenom

Having Count(Distinct e.ID\_stage) >= 2;

Réponses aux 4 requêtes

1) Lister les stages qui sont encore ouverts.

id_stage	intitule_stage
1000	Cybersécurité
1002	Intelligence Artificielle
1004	Systèmes embarqués
1006	Sécurité des réseaux
1008	Applications mobiles
1010	Blockchain
1012	DevOps
1013	Machine Learning

ce que  
fait chaque  
question

zone de  
résultat

Question 1      Question 2      Question 3      Question 4

cliquer pour  
naviguer entre  
les questions