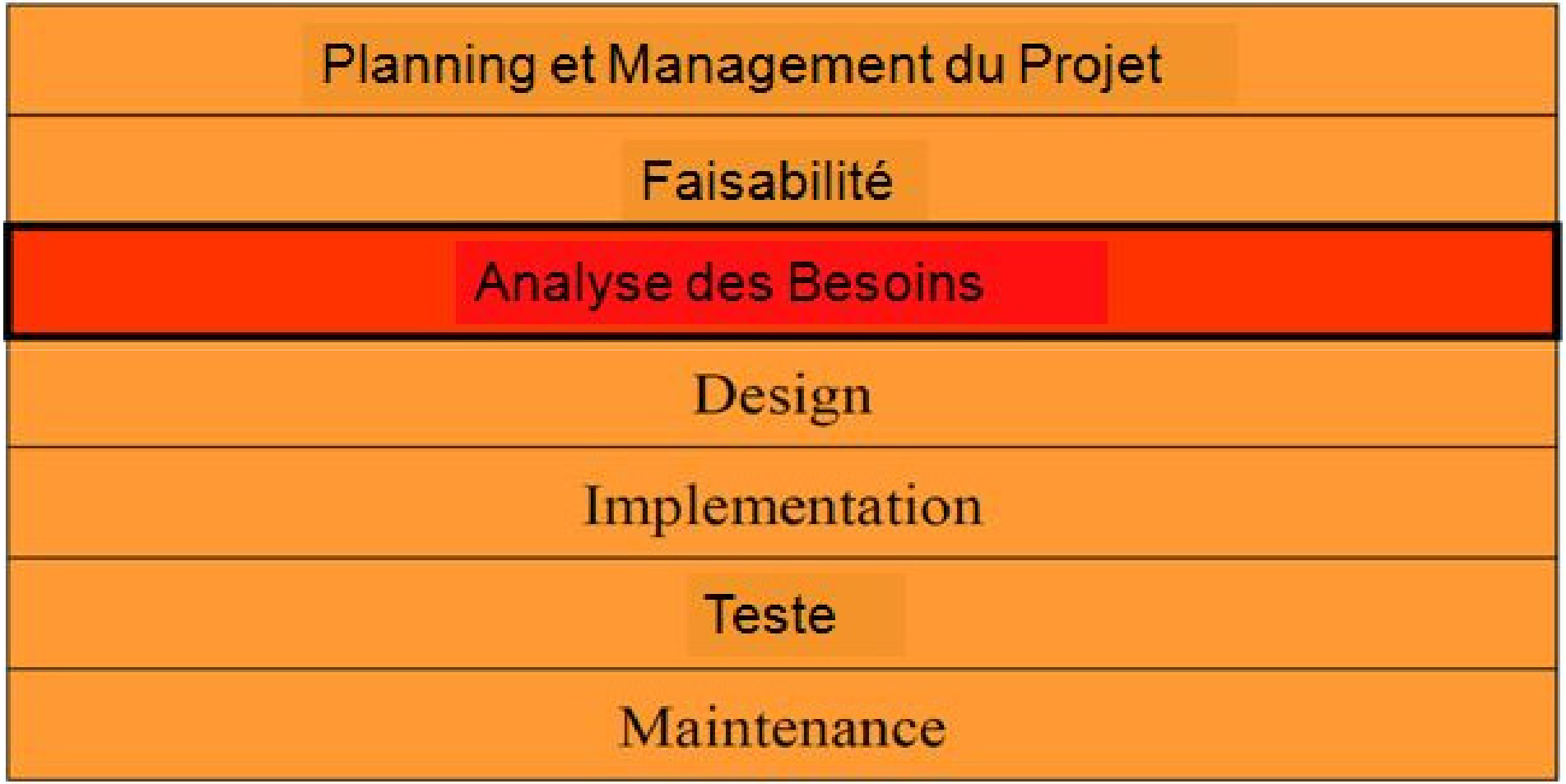
2. Database Design

www.LearnDB.com

Dr. Imed Bouchrika
Dept of Mathematics & Computer Science
University of Souk-Ahras
imed@imed.ws

Thanks are due to Nishadha & Griffin for their notes

Software Lifecycle



Database design lifecycle

- **Analyse de Besoins**
  - Les besoins du clients; ce que la base de donneés doit faire?
- Conception
  - Une description haute niveau; souvent utilisé le modele E/R
- Conception Logique
  - Transformer le modèle E/R (typiquement) en une schéma relationnelle.
- Refinement du schema
  - Vérifier les redandance et les anomalies.
- Conception physique/accordement
  - Considerer la quantité du travail, et faire plus d'optimisation.

Analyse de Besoins

- Durant cette phase, on a besoin de savoir le plus maximum possible sur :
  - Les utilisateurs.
  - Les Taches/Fonctionnalitées.
  - Le Contexte.

- Pour qu'on peut identifier les types des utilisateurs et leurs besoins.

- Cette phase est extrémement IMPORTANTE car plus de 50% des érreurs/ méprise du logiciel sont causé par des érreurs faites au niveau de cette phase.

Analyse de Besoins

- Problèmes qui s'imposent au Analyse de Besoins:
  - Les utilisateurs ne savent pas ce qu'ils veulent!
  - Les utilisateurs supposent que TU sache tout.
  - Les Besoins se different d'une personne à une autre.
  - Les Facteurs Politiques et Sociales !
  - Les choses se changent rapidement.

Analyse de Besoins

- Les Spécifications Fonctionnelles
  - Ce que l'utilisateur veut faire.
  - Ce que l'utilisateur veut voir.

- Les Spécifications non Fonctionnelles
  - Securité
  - Efficacité
  - Performance
  - Disponibilité
  - Facilité d'utilisation.
  - ....

Analyse des Besoins

- Activité de Groupe : Essayer de produire un ensemble des besoins pour les systèmes des logiciels suivant:
  - Bibliothèque Personelle.
  - Bibliothèque de l'Université.
  - Résérvation dans un Hotel.
  - Un petit moteur de rechereche.

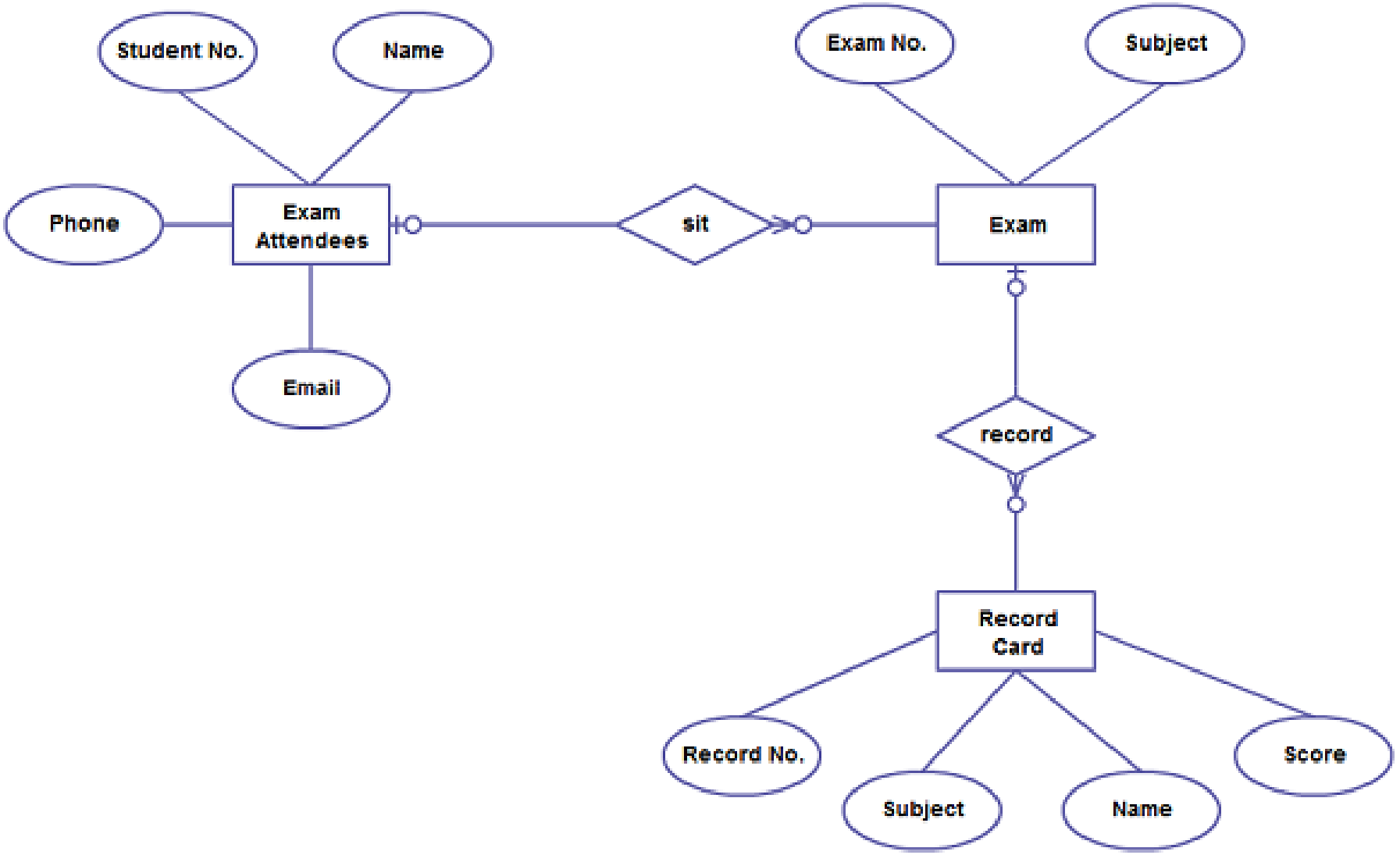
Cycle de vie du design de la Base de Données

- Analyse des Besoins
  - Les besoins du clients; ce que la base de donneés doit faire?
- **Conception**
  - Une description haute niveau; souvent utilisé le modele E/R
- Conception Logique
  - Transformer le modele E/R (typiquement) en une schéma relationnelle.
- Refinement du schema
  - Vérifier les redandance et les anomalies.
- Conception physique/accordement
  - Considerer la quantité du travail, et faire plus d'optimisation.

Diagramme d'Entité-Relation (DER)

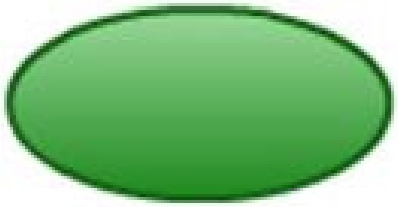
- DER est un language graphique utilisé pour les design des base de données.
- DER est utilisé pour fournir une representation graphique pour la structure logique de la base de données.
- Ils sont facile à comprendre et ne demandent pas un entrainement étendu.
- DER peut etre utilisé pour faciliter la communication avec les developpeurs, clients et utilisateurs.
- DER sont traduisables en tables relationnelles qui peut etre utilisé pour facilement construire la base de données.

Diagramme d'Entité-Relation (DER)

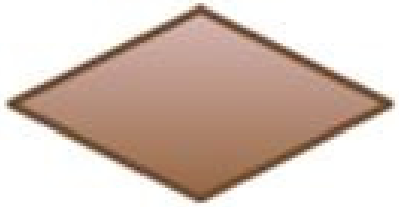


Notions d'Entité-Relation (E/R)

Entité



Attribut



Relation

Entité
Faible

Attribut
Multi-valeurs

Relation
Faible

Notions d'Entité-Relation (E/R)

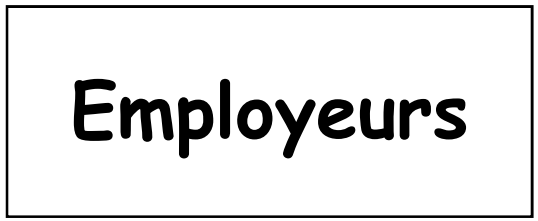
- Une **entité** est un objet du monde réel qui est differents des autres objets.

- Une entité peut etre une personne, un endroit, un événement ou n'importe quelle chose qui possède des propriétés.

- Exemple, un système d'école peut inclure :
  - Des étudiants,
  - Des enseignents,
  - Des coures,
  - Des modules ....

Entité et attributs

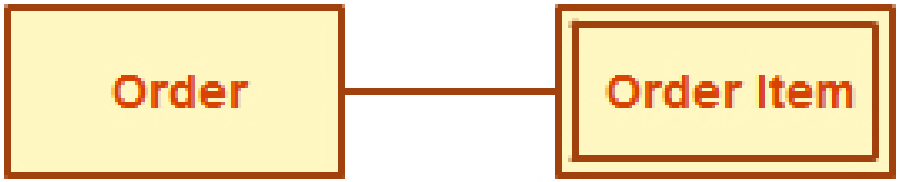
- **l'Entité** est représenté par un ***rectangle*** .



- Chaque Entité possède des **attributs** .

Entité Faible

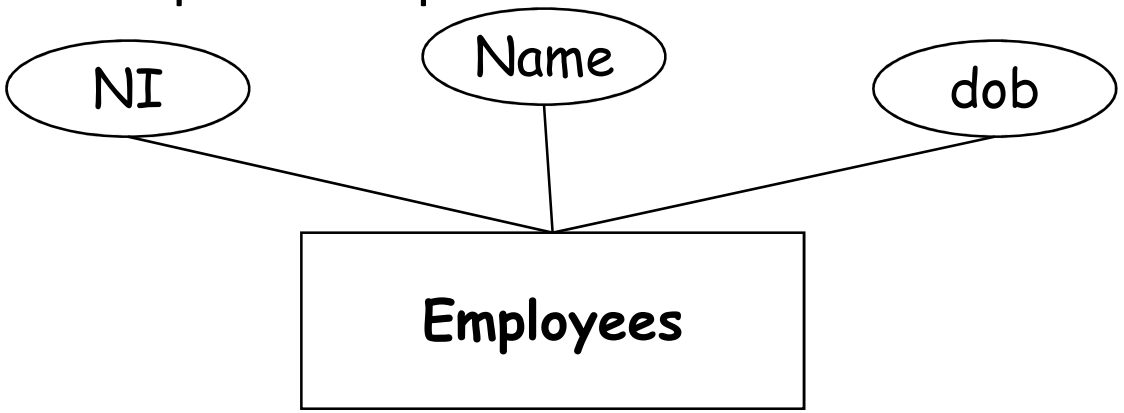
- **L'entité Faible** est une entité qui se **depend** sue **l'existance** d'une autre entité.
- *Elle* ne peut pas etre identifier par ses propre attributs.
- Elle utilise le clé étrangère combiner avec ses attributs pour former un clé primère.
- L'entité faible est représenté par un **rectangle** double ligné.



Notions d'Entité-Relation (E/R)

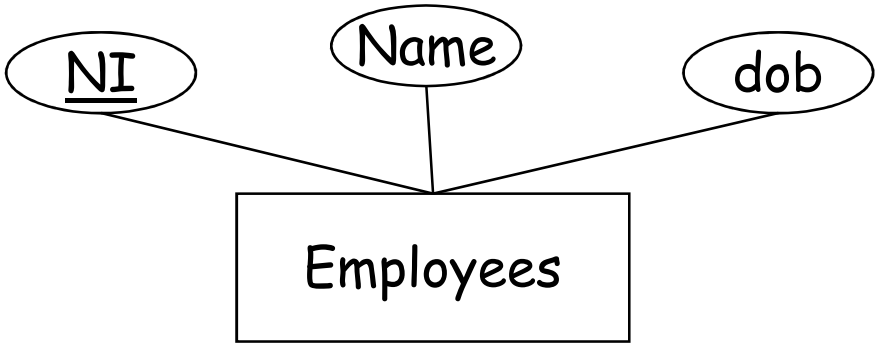
- Un **attribut** est une propriété, un trait ou caracthéristique de:
  - Une entité.
  - Une relation.
  - Un autre attribut.

- Un **Attributs** est représenté par un ***ovale.***



L’attribut clé

- Un **attribut clé** pour une entité est un attribut dont sa valeur est **differente** pour chaque entité.



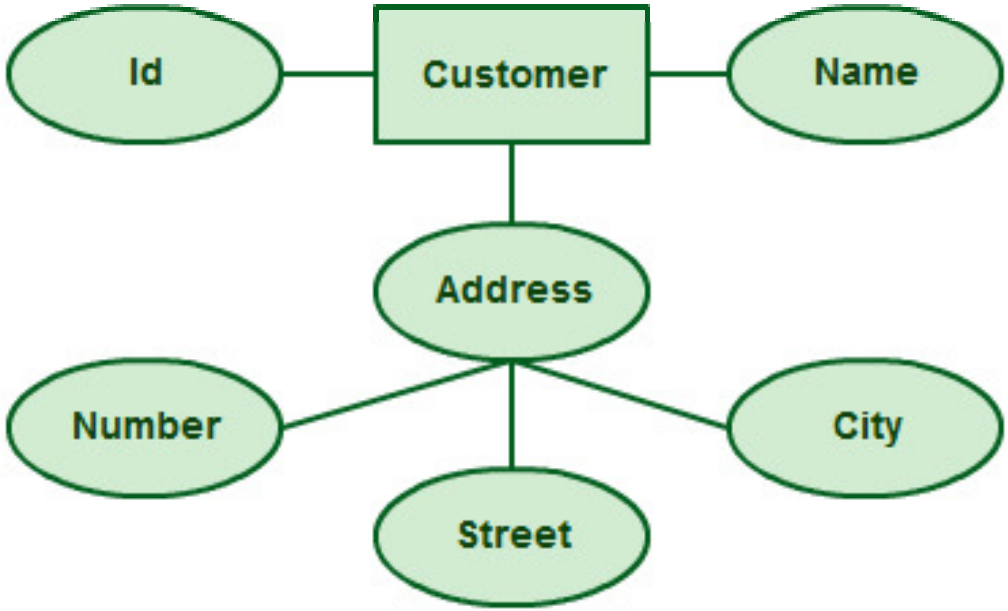
- On **souligne** l’attributs clé.

- **NI** est l’attributs clé dans l’exemple au-dessous.

- Parfois plusieurs attributs ensemble forme une clé.
  - NB: on doit prendre le nombre **minimale** des attributs.

Composite attributes

- Les attributs aussi peuvent avoir **ses propres** spécifique attributs.
- Exemple: l'attribut "customer address" peut avoir les attributs : *number*, *street*, *city*, and *state*.
- Ils sont appelé les **attributs composite**.



Attribut Multi-valeur

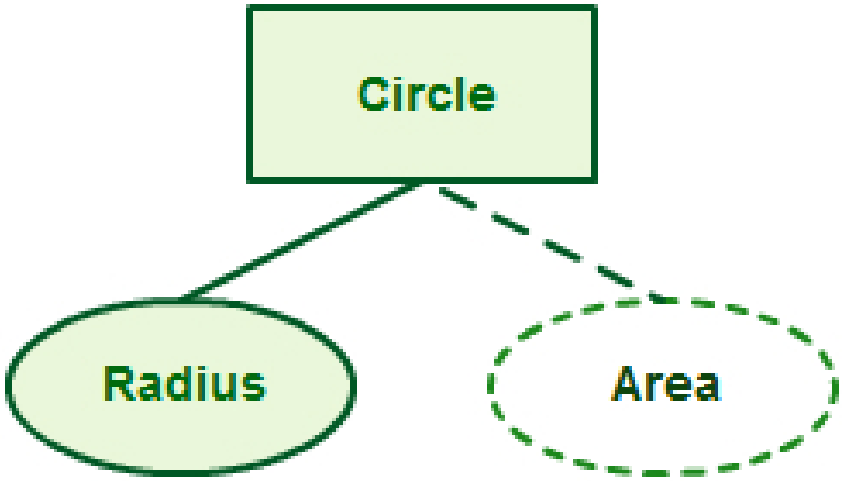
- Si une entité peut avoir plusieurs attributs, on utilise un **ovale double ligné** pour l'attribut.

- Exemple: l'entité enseignent peut avoir plusieurs valeurs de module.

L'Attribut dérivé

- L'Attribut dérivé : peut etre calculer à partir d'un autre attribut.

- Exemple: la surface du cercle peut etre dérivé à partir du rayon.



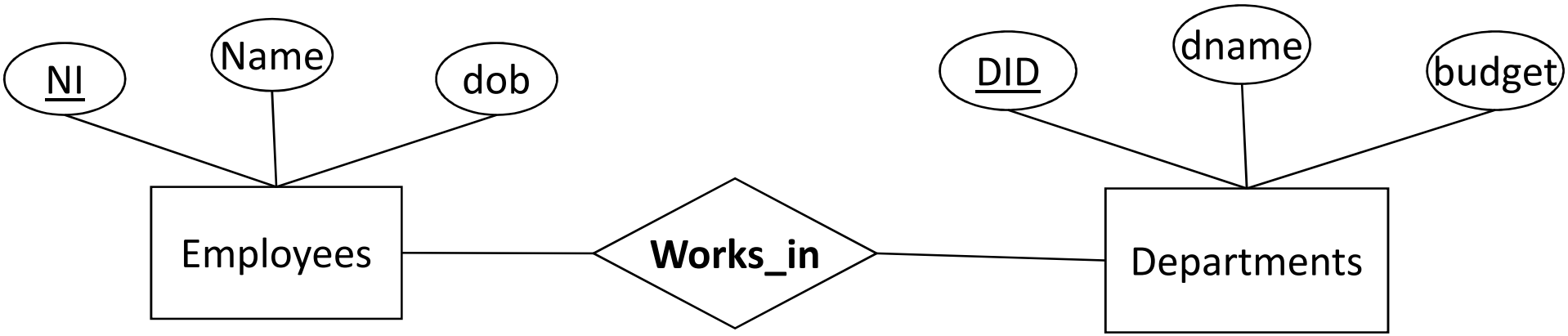
Les Relations en E/R

- Une **relation** décrit comment les entités intér gissent entre eux.

- Les types de Relations sont représenté par un ***diamond.***

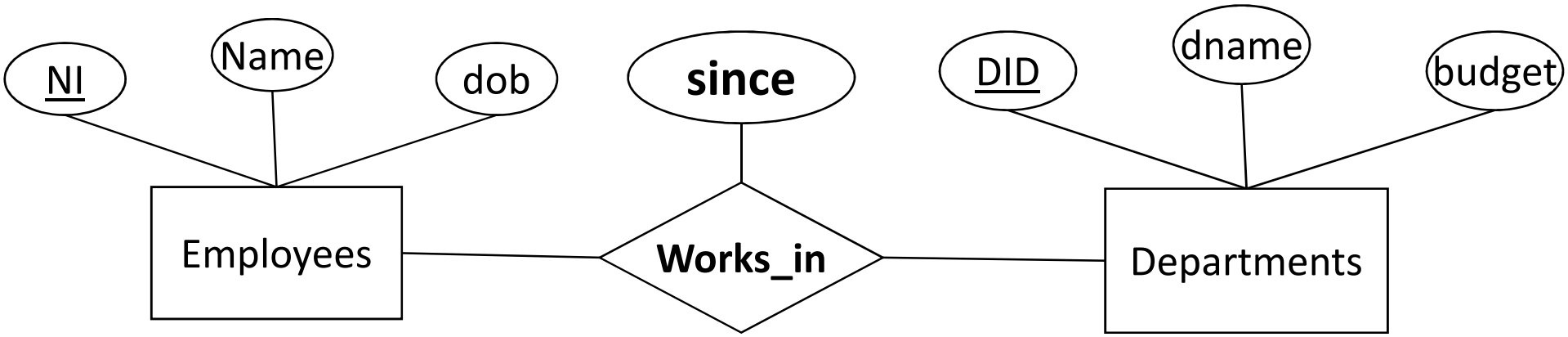
- Ils relient les entités participante par une **ligne**.

Les Relations en E/R



Les Attributs d'une Relation

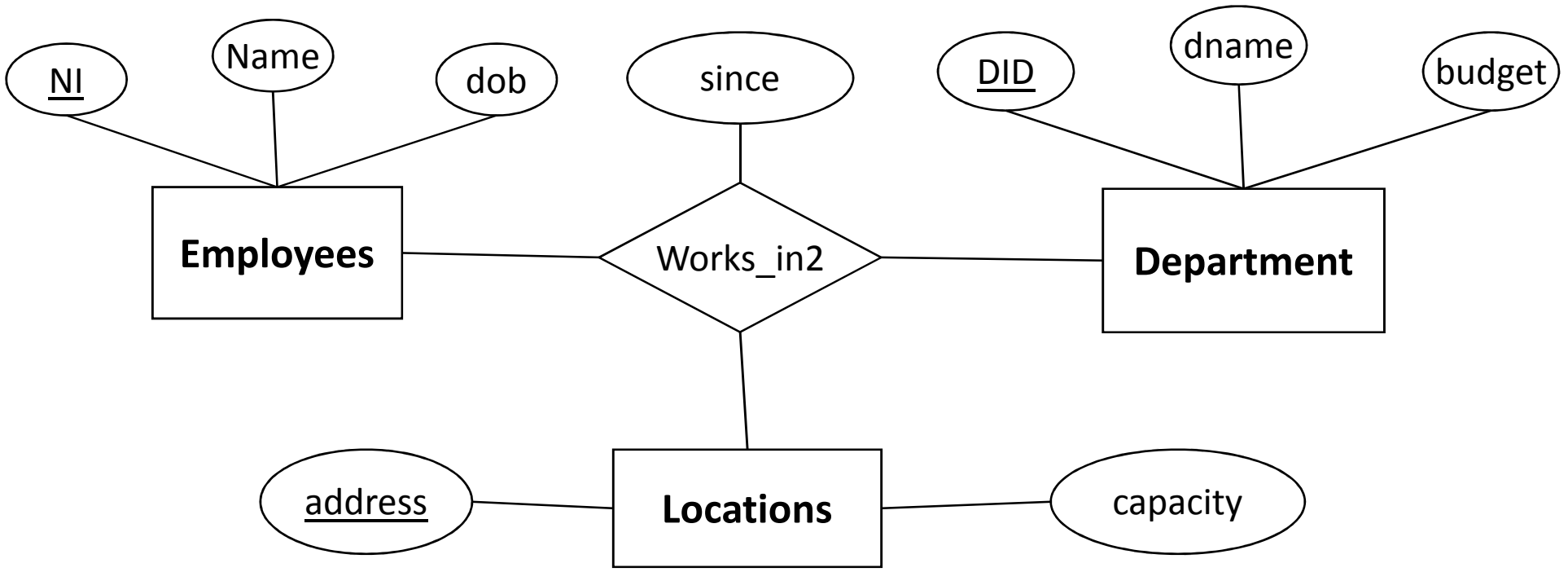
- les Relations aussi peuvent avoir des **attributs**



Les relations N-aire

- On peut avoir une relation **n-aire** en plus que binaire.

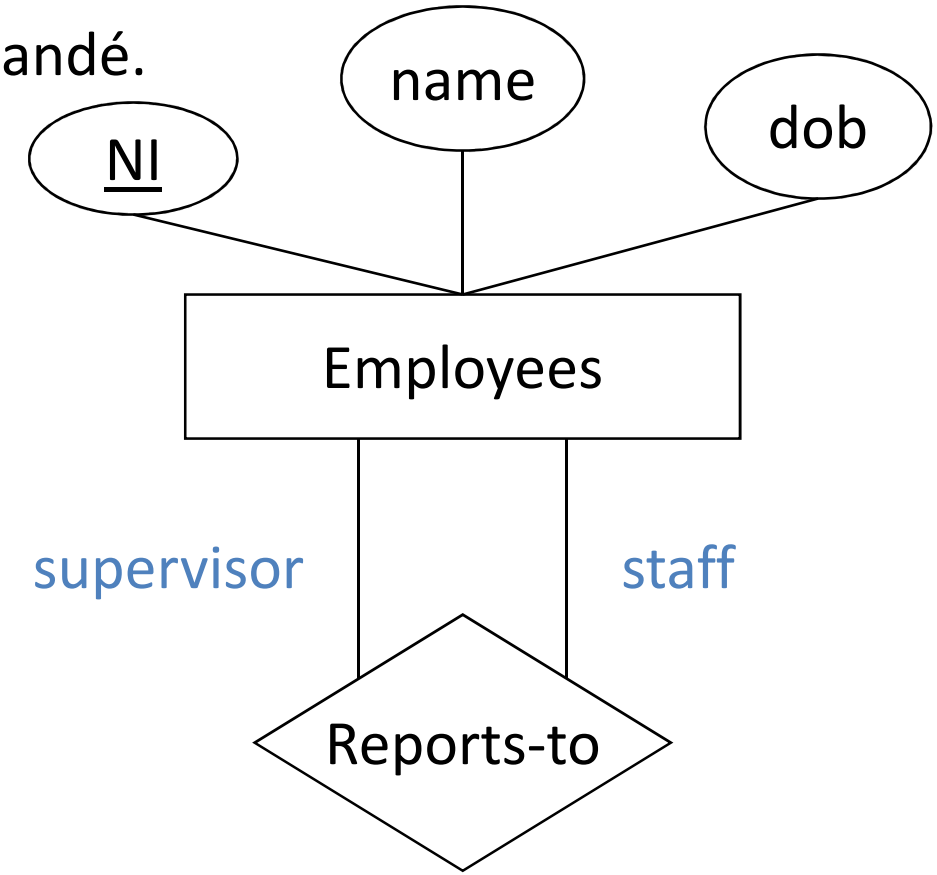
- Un exemple d'une relation **Ternaire**.



Les Relations Recursive

- **Une relation Recursive** c'est quand une entité joue plusieurs roles dans la relation.

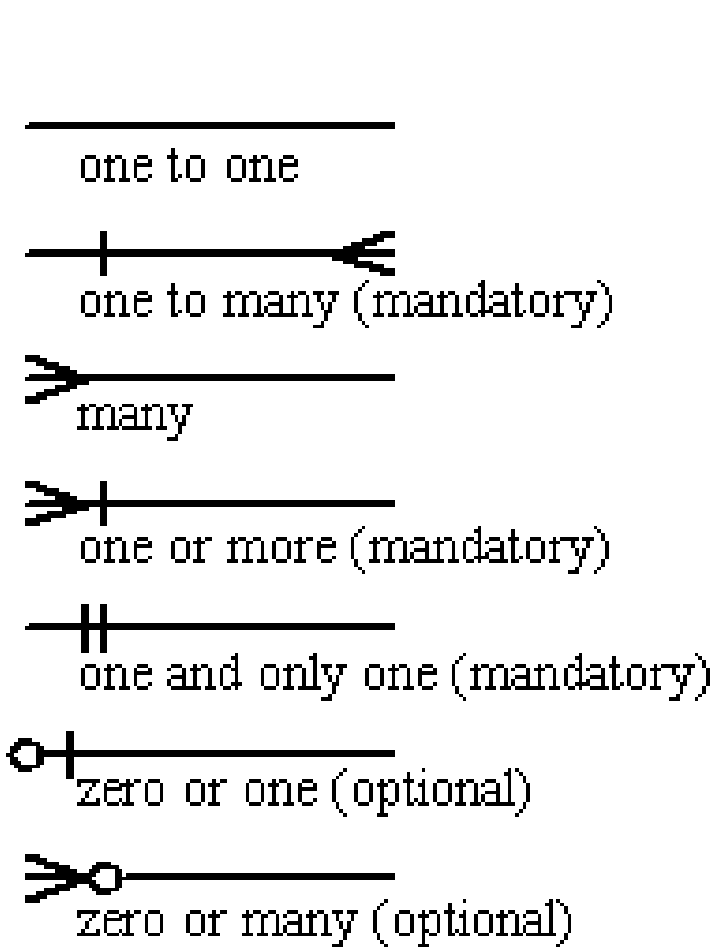
- ***Le nom du role*** est demandé.

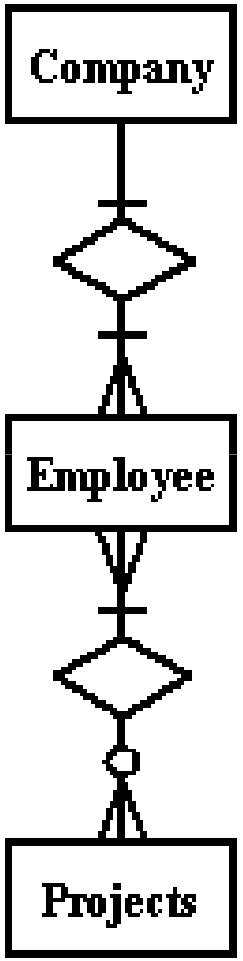


Les Cardinalité

- Exemple:
  - Un employeur peut travailler dans plusieurs departments; un department peut avoir plusieurs employeurs.
  - En opposition, chaque department a un seul manager.
- Les rapports possible sont: **1:1**, **1:N**, **N:1**, **M:N**

Les Cardinalité





Les contraintes de Participation

Chaque department doit avoir un manager

- C'est un exemple de **contrainte de participation.**

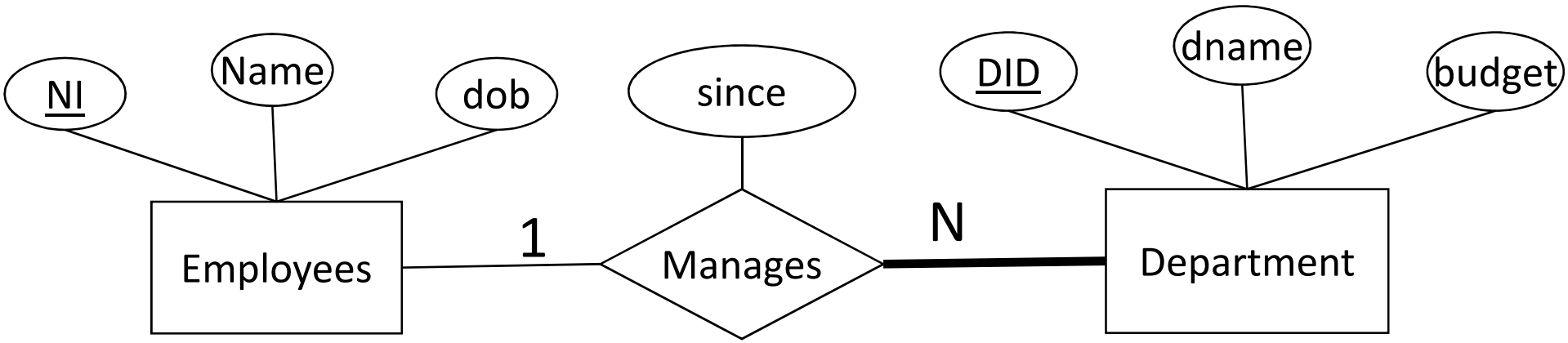
- La participation d'un ensemble d'entité, E, dans un ensemble de relations, R, est dit **total** si:

  → chaque entité de l'ensemble E participes **au moins dans une** relation de l'ensemble R.

- Sinon, on dit que c'est une participation **partielle.**

La Participation dans les diagrammes d'E/R

- La participation totale est représenté par une ligne **gras** entre l'entité et la relation.
  - NB. Parfois on le trouve écrite: **double ligne.**



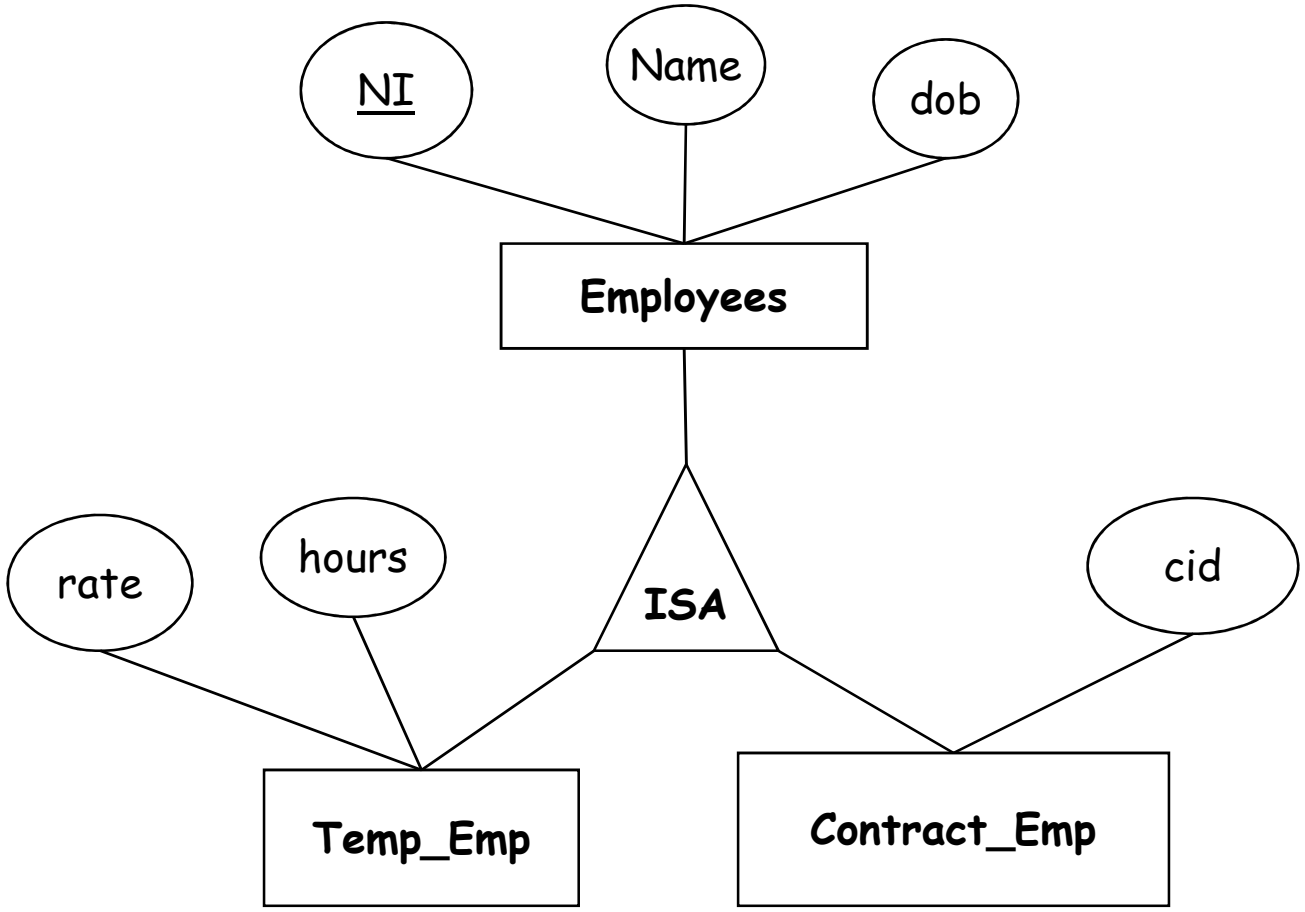
La modélisation étendu de l'E/R (DERE)

- Durant les années, un nombre de caracthéristique a été ajouté au modèles et à la processus de modélisation.
- Ces caracthéristiques inclut:

  - Sous et super-classes
  - Specialisation
  - Généralisation
  - Categoris
  - haut/ bas-niveau ensemble d'entité
  - Héritage des attributs
  - Aggrégation

ISA hierarchies

- On peut concevoir des **hierarchies** pour les entités.



Héritage des Attributs

- Les ***attributs*** du superclasses sont hérité par la sousclasses.
- Ainsi: Temp_Emp aussi possède les attributs NI, Name and dob.
- Les sousclasses héritent les ***relations*** aussi.

De L'ERD vers L'SQL

- Entités → Table
- Attributs → Columns
- Attribut clé → Primary Key
- Entité Faible → Use Foreign Key to refer to their Parent Entity
- Relations → Table

UML

- *The Unified Modeling Language (UML) est un language graphique pour communiquer les spécifications du design d'un software.*

- *Les Diagrammes d'UML:*
  - *Diagrammes de Structure*
    - *Classe, Composant, Deploiement ...*
  - *Diagrammes de Comportement*
    - *Activité, état, cas d'utilisation*
  - *Diagrammes d'Intéraction*
    - *Séquence, Communication ...*

Étude de cas: Youtube

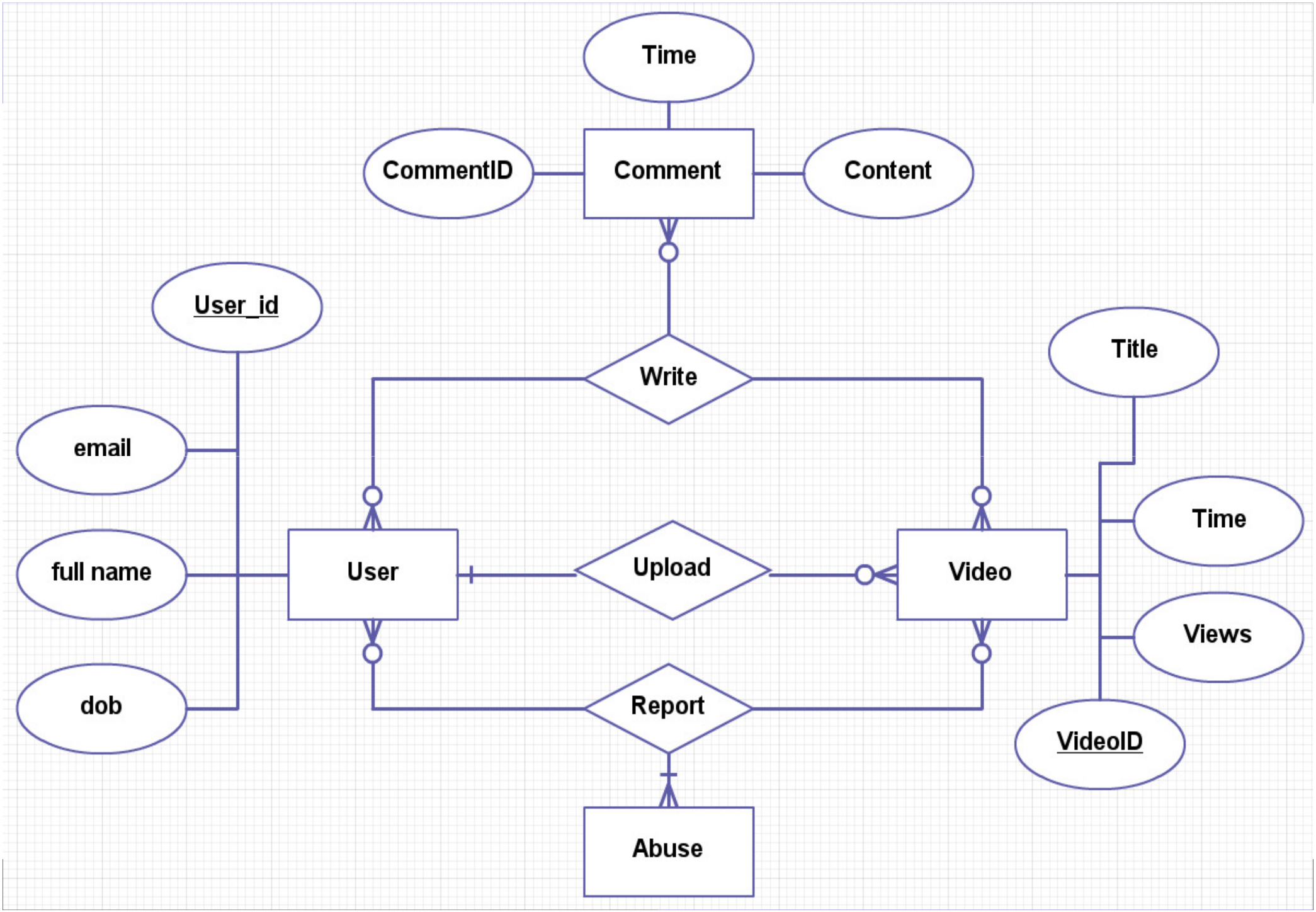
- ***Décrit les fonctionalités les plus demandé:***
  - Un utilisateur peut transférer une Video
  - Le personnel de Youtube peut controlé les videos pour les copyright ...
  - Le personnel de Youtube peut supprimé des videos.
  - Youtube peut bloqué des utilisateurs.
  - Les Visiteurs peuvent regarder la video online.
  - Les Videos doivent avoir un compteur.
  - Un Visiteur peut accédé à une session ou inscrive.
  - Les utilisateur online peuvent commenté sur une video.
  - Les utilisateur online peuvent signalé des videos.
  - ……

Étude de cas: Youtube

- ***Décrit les fonctionalités les plus demandé:***
  - Un **utilisateur** peut transférer une **Video**
  - Le **personnel** de Youtube peut controlé les videos pour les copyright ...
  - Le personnel de Youtube peut **supprimé** des videos.
  - Youtube peut **bloqué** des utilisateurs.
  - Les Visiteurs peuvent regarder la video online.
  - Les Videos doivent avoir un **compteur.**
  - Un Visiteur peut accédé à une session ou inscrive.
  - Les utilisateur online peuvent **commenté** sur une video.
  - Les utilisateur online peuvent **signalé** des videos.
  - .....

Étude de cas: Youtube

- ***Les entités possibles sont:***
  - Utilisateur
  - Video
  - Personnel
  - Commenté
  - Signalé
  - Bloqué un utilisateur


Time
CommentID
Comment
Content
User_id
Write
Title
email
full name
dob
User
Upload
Video
Time
Views
Report
VideoID
Abuse

À vous de savoir !

- **Database Normalisation .**
- **Chen Notation.**
- ***Software Artifact***