

---

# Technologies du commerce électronique

Travail pratique #01: partie 01

---

# TP #01: partie 01

---

## ▶ Objectifs

- ▶ Se familiariser avec le développement d'applications Web avec Ruby On Rails
- ▶ Apprendre à utiliser:
  - ▶ Le router
  - ▶ Les migrations
  - ▶ La validation de données
  - ▶ Les vues et les formulaires imbriqués
  - ▶ Les contrôleurs
  - ▶ ...
- ▶ Rendu graphique avec Bootstrap ou MDB

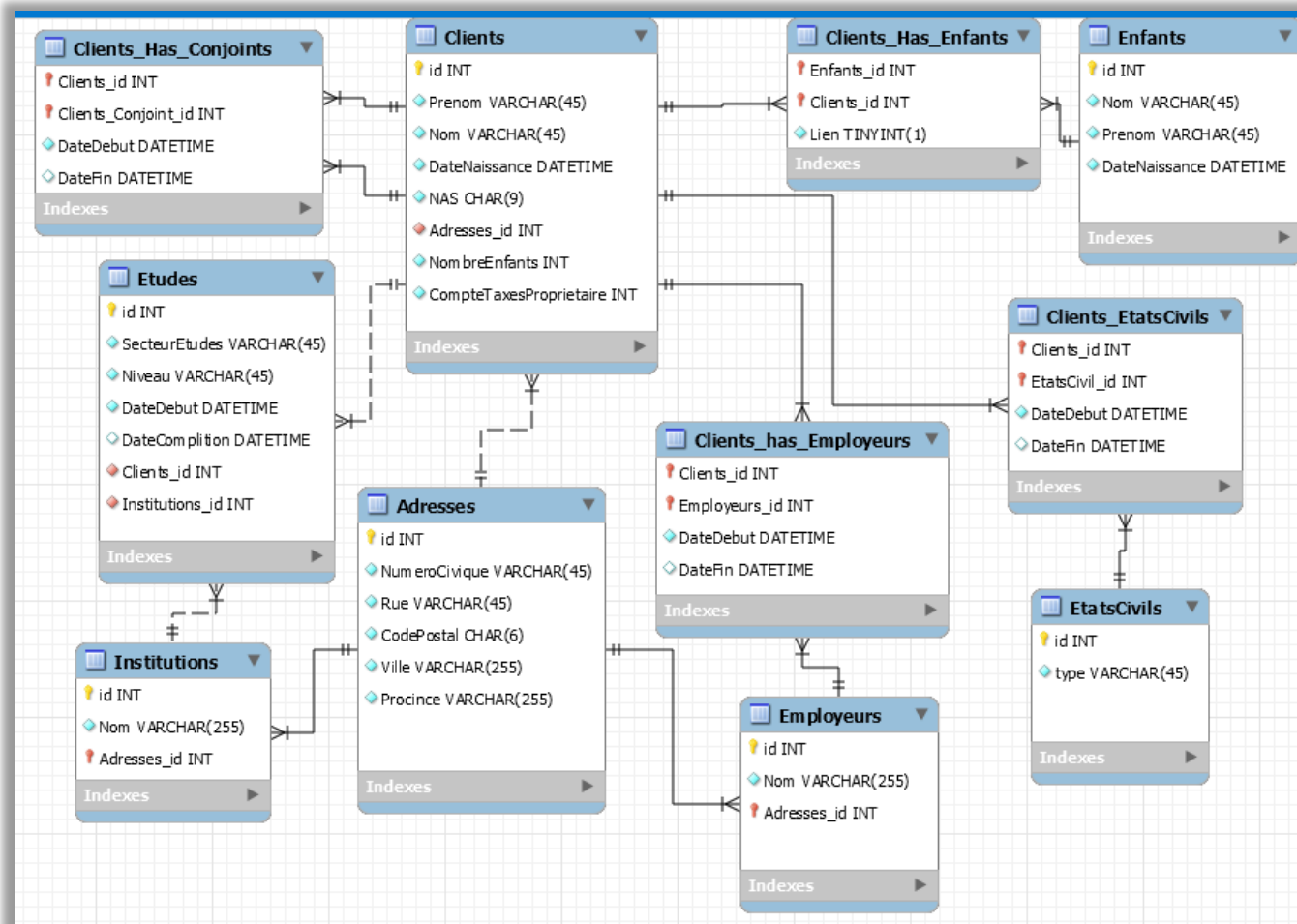
# Objectifs

---

- ▶ Développement d'une application Ruby On Rails qui permet d'effectuer des opérations CRUD sur un formulaire “complexe”
- ▶ Bonus:
  - ▶ Développement d'interface utilisateur avec MDB

# Énoncé: modèle relationnel

- Capture d'écran du modèle relationnel (fichier MySQLWorkbench est disponible sur le site du cours)



Remarque: si nécessaire, vous pouvez adapter ce modèle

# Étapes

---

- ▶ Générer les fichiers de migrations pour la création des différentes tables
- ▶ Création des contrôleurs et des vues pour pouvoir effectuer des opérations **CRUD** sur les modèles
  - ▶ Création d'un « grand » formulaire (pour le contrôleur associé au modèle Clients) qui imbrique les différents modèles associés
    - ▶ Indication: utiliser « nested forms »:
      - ▶ [http://guides.rubyonrails.org/form\\_helpers.html#nested-forms](http://guides.rubyonrails.org/form_helpers.html#nested-forms)
      - ▶ [https://github.com/plataformatec/simple\\_form/wiki/Nested-Models](https://github.com/plataformatec/simple_form/wiki/Nested-Models)
- ▶ Rendu graphique: utiliser Bootstrap 4.0
  - ▶ <https://getbootstrap.com/docs/4.0/getting-started/download/>
- ▶ Adapter l'interface utilisateur pour intégrer MDB (BONUS):
  - ▶ <https://mdbootstrap.com/material-design-for-bootstrap/>

# Directives

---

- ▶ Travail à effectuer en équipes de 2 personnes
  - ▶ Peut tolérer des équipes de 3 personnes (dans ce cas, le point BONUS deviant obligatoire)
- ▶ Pour la gestion des dépôts, git sera utilisé et l'historique doit faire ressortir la participation effective et régulière de chaque membre d'une équipe
  - ▶ Les messages (commentaires) des *commits* doivent être suffisamment détaillés, explicites et significatifs
- ▶ Fichiers à remettre:
  - ▶ Code source associé au projet sous forme d'un dépôt Git
  - ▶ Fichier texte, word ou markdown ou présentation powerpoint expliquant votre démarche, les problèmes rencontrés et les solutions proposées
- ▶ Remise via le site du cours au plus tard le 18 février 2018 à 10h
- ▶ Une séance serait éventuellement programmée pour que chaque équipe puisse présenter son travail
- ▶ Travail à réaliser en utilisant Ruby On Rails avec Sqlite comme SGBDR

# Évaluation

---

- ▶ Les éléments suivants seront pris en considération lors de l'évaluation de ce travail:
  - ▶ Complétude
  - ▶ Qualité du code
  - ▶ Qualité des interfaces graphiques
  - ▶ Couverture minimal en tests (unitaires, intégration)
  - ▶ Apport et implication de chaque coéquipier (le code du projet doit être un dépôt git qui inclus l'historique des modifications )
  - ▶ Capacité à aller chercher et retrouver l'information pertinente
  - ▶ Respect des consignes