

## Exercices de Modélisation et d'Algèbre Relationnelle L1IRT 2025

### Partie 1 : Exercices de Modélisation

Pour chaque exercice, faites les trois formes normales et réalisez le MCD avec le MLDR.

1. Modéliser une base de données pour une bibliothèque (livres, auteurs, emprunts, membres).

Relation(ISBN, titre, date\_publication, nb\_pages, id\_auteur, id\_auteur, nom, prénom, pays, id\_membre, nom, prénom, date\_inscription, email, id\_emprunt, date\_emprunt, date\_retour)

Un livre est emprunté par un membre à une date donnée et l'emprunt à une date de retour. Un Livre ne peut être emprunté que s'il existe un Membre et que le stock du Livre est  $\geq 1$ . Un Membre ne peut avoir plus de 5 Emprunts simultanés. Un Emprunt doit toujours référencer un Livre et un Membre existants.

2. Concevoir un schéma relationnel pour un système de gestion des étudiants et des cours.

Relation(id\_étudiant, nom, prénom, date\_naissance, email, id\_cours, nom\_cours, crédits, date\_inscription, note)

Un Étudiant ne peut s'inscrire à un Cours que si le cours existe. Une Inscription lie exactement un Étudiant et un Cours, et la date\_inscription doit être dans la période d'ouverture des inscriptions. La note d'une Inscription est toujours comprise entre 0 et 20.

3. Modéliser une base de données pour un hôpital (patients, médecins, consultations, traitements).

Relation(id\_patient, nom, prénom, date\_naissance, adresse, téléphone, id\_médecin, nom, spécialité, téléphone, id\_consultation, date\_consultation, motif, id\_traitement, description, durée\_jours)

Un Patient ne peut avoir qu'une seule consultation par médecin et par heure. Un médecin ne peut planifier plus de 8 consultations par jour. Tout Traitement doit référencer une Consultation existante. Un Patient ne peut être supprimé si des consultations ou traitements l'impliquent.

4. Créer un modèle relationnel pour un site de commerce électronique (produits, clients, commandes, paiements).

Relation(id\_produit, nom, description, prix, stock, id\_client, nom, prénom, email, adresse, id\_commande, date\_commande, montant\_total, quantité, prix\_unitaire, id\_paiement, date\_paiement, montant, mode\_paiement)

Le prix unitaire d'un produit est défini lors de la commande et ne peut-être inférieur au prix du produit. Un paiement est fait par rapport à une commande faite préalablement.

Une Commande ne peut être validée si le stock de l'un des Produits est insuffisant. Le montant\_total de la Commande =  $\Sigma$  (quantité  $\times$  prix\_unitaire). Un Paiement doit couvrir intégralement le montant\_total. Un Client ne peut passer une nouvelle commande s'il a un paiement en retard de plus de 15 jours.

5. Modéliser une base pour une entreprise de transport (bus, chauffeurs, trajets, Transports).

Relation(id\_bus, immatriculation, capacité, id\_chauffeur, nom, prénom, permis\_numéro, id\_trajet, origine, destination, distance\_km, date\_heure\_depart, date\_heure\_arrivee)

Un Chauffeur ne peut conduire plus de 10 heures par jour. Les horaires d'arrivée doivent être après l'heure de départ. Un Trajet doit avoir au moins deux horaires planifiés pour le transport par jour. Un transport est faite par un chauffeur sur trajet avec un bus avec une date heure de départ.

6. Concevoir une base pour une agence de voyage (destinations, clients, réservations, vols).

Relation(id\_dest, ville, pays, description, id\_client, nom, prenom, email, id\_vol, compagnie, date\_heure, id\_resa, date\_réservation, statut, id\_billet, siège)

Un vol est un déplacement entre deux destinations, soit celle de départ et celle d'arrivée. Un billet émis est relié à une réservation et un vol. On enregistre le siège concerné. Une Réservation ne peut être confirmée que si le vol et la destination existent. Nombre maximum de sièges par Vol = capacité fixée par la compagnie. Un Billet doit être émis dans les 24 h suivant la confirmation de la Réservation. Le statut 'Annulée' d'une Réservation entraîne l'invalidation de son Billet.

7. Modéliser une base de données pour un réseau social (utilisateurs, publications, commentaires, amis).

Relation(id\_user, pseudo, nom, prénom, date\_naissance, email, id\_pub, contenu, date\_publication, id\_com, contenu, date\_commentaire, date\_amis)

Deux utilisateurs deviennent amis à une date donnée. La relation d'amitié est symétrique : si A est ami de B, alors B est ami de A. Une publication est faite par un

utilisateur. Un commentaire est fait pour une publication et par un utilisateur. Un commentaire peut faire office d'un autre commentaire. Un Utilisateur ne peut commenter qu'après avoir publié au moins une Publication. Un Commentaire ne peut être posté plus de 48 h après la Publication d'origine. Un Utilisateur ne peut avoir plus de 5 000 amis.

8. Créer un modèle pour une école (élèves, classes, enseignants, matières).

Relation(id\_eleve, nom\_eleve, prenom\_eleve, date\_naissance, id\_classe, niveau, section\_classe, id\_ens, nom\_ens, specialite\_ens, email\_ens, id\_mat, nom\_matiere, coefficient)

Un Élève ne peut appartenir qu'à une seule Classe. Un Enseignant ne peut enseigner plus de 5 matières différentes. Un enseignement est fait par un enseignant pour une classe dans une matière. Une Affectation Enseignement doit respecter les créneaux de la Classe. Les coefficients doivent être compris entre 0,5 et 5.

9. Modéliser un système de gestion des stocks pour un entrepôt (articles, fournisseurs, livraisons, commandes).

Relation(id\_article, nom\_article, description, prix\_unite, quantite\_stock, id\_fournisseur, nom\_fournisseur, pays\_fournisseur, contact\_fournisseur, id\_livraison, date\_livraison, id\_commande, date\_commande, statut\_commande, quantite)

Une commande est faite vers un fournisseur. Une livraison est faite pour une commande. Pour chaque article sur une commande on enregistre la quantité commandée. Le stock ne peut jamais devenir négatif. Chaque Livraison doit arriver dans un délai maximal de 15 jours ouvrés après la date de commande fournisseur. Une Commande client ne peut être expédiée que si tous les articles sont en stock. Le prix\_unité peut varier d'au plus  $\pm 10\%$  par an sans approbation.

10. Concevoir une base de données pour une salle de sport (adhérents, abonnements, cours, entraîneurs).

Relation(id\_adherent, nom, prenom, date\_naissance, email, id\_abo, type\_abo, date\_debut, date\_fin, id\_cours, nom\_cours, niveau\_cours, capacite\_cours, id\_ent, nom\_ent, specialite\_ent, id\_plan PK, date\_heure)

Un cours est planifié par un entraîneur à une heure donnée. Un abonnement est fait par un adhérent. Un adhérent peut suivre plusieurs cours planifiés tant qu'il y a un abonnement en cours et un cours planifié est suivi par plusieurs adhérents. Le nombre total d'adhérent ayant suivi un cours planifié ne peut dépasser la capacité du cours. Un Adhérent ne peut avoir qu'un abonnement actif à la fois. Un Entraîneur ne peut animer

plus de 3 cours simultanément.

11. Modéliser une base pour un cinéma (films, séances, salles, réservations).

Relation(id\_film, titre, réalisateur, durée\_min, genre, id\_salle, numero\_salle, capacité, id\_séance, date\_heure, prix, id\_reser, nb\_places)

Une Séance ne peut être programmée qu'avec un Film existant et une Salle disponible. Une Salle ne peut accueillir deux Séances dont les créneaux se chevauchent. Le prix de la Réservation = nb\_places × prix de la Séance. Une Réservation doit être confirmée au moins 2 heures avant la Séance. La somme des nombre de places réservées pour une séance ne peut dépasser la capacité de la salle.

12. Créer un modèle relationnel pour une université (départements, enseignants, cours, étudiants).

Relation(id\_dep, nom\_dep, bâtiment, id\_cours, titre, crédits, id\_étudiant, nom\_etu, prenom\_etu, date\_inscription, semestre, note, date\_composition)

Un étudiant appartient à un seul département. Un étudiant compose une fois pour un cours dans un semestre et obtient une note. Un cours peut être fait dans plusieurs départements et un département à plusieurs cours mais pour un cours dans un département on enregistre le crédit associé et l'enseignant qui l'enseigne. Un Étudiant ne peut composer dans un Cours si son Département n'est pas autorisé par le Département du Cours. Un Cours doit offrir au moins 3 crédits. Un Enseignant ne peut dispenser plus de 4 cours par semestre. La somme des crédits de cours par semestre dans un département doit faire 60 crédits.

13. Modéliser une base pour une entreprise de location de voitures (véhicules, clients, locations, entretiens).

Relation(id\_vehicule, marque, modèle, année, immatriculation, id\_client, nom, prenom, permis\_numero, téléphone, id\_location, date\_début, date\_fin, tarif\_journalier, id\_ent, date\_entretien, type, coût)

Un véhicule se fait entretenir de temps en temps avec un coût, un type à une date.

Une location est faite par un client pour un véhicule. Une Location ne peut démarrer que si le Véhicule est disponible et sans entretien en cours. Le tarif\_journalier dépend du type et de l'âge du Véhicule. Un Entretien doit être planifié tous les 10 000 km ou tous les 6 mois (selon le premier terme atteint). Le coût d'entretien est ajouté à la facture finale du Client si hors garantie.

14. Concevoir un système de gestion de projets (projets, employés, tâches, délais).

Relation(id\_projet, nom\_projet, date\_début, date\_fin\_est, id\_emp, nom, prénom, poste, id\_tâche, description, date\_échéance, statut, date\_affect)

Une tâche est en fonction d'un projet. Un employé se fait affecter une tâche à une date donnée. Une Tâche ne peut être créée sans  $\text{date\_échéance} \geq \text{date\_début}$  du Projet. Un Employé ne peut avoir plus de 5 Tâches simultanées. La  $\text{date\_fin\_est}$  du Projet =  $\max(\text{date\_échéance des Tâches}) + \text{marge}$ . L'affectation est effective seulement si la Tâche n'est pas encore terminée.

15. Modéliser une base de données pour un restaurant (plats, menus, commandes, clients, serveurs).

Relation(id\_plat, nom\_plat, prix\_plat, catégorie, id\_menu, nom\_menu, prix\_menu, id\_client, nom\_cli, prenom\_cli, téléphone, id\_srv, nom\_srv, prenom\_srv, id\_comm, date\_heure)

Un menu contient plusieurs plats qui peuvent chacun se retrouver dans un autre menu. Une commande est faite par un client et est servie par un serveur à une heure donnée. Une commande peut contenir plusieurs plats mais pour chaque plat on enregistre la quantité commandée. Un Menu ne peut inclure qu'au maximum 8 Plats. Le prix du Menu =  $\sum \text{prix des Plats} - \text{remise éventuelle}$ . Une Commande doit être transmise au Serveur en moins de 2 minutes après prise par le Client.

## Partie 2 : Exercices d'Algèbre Relationnelle (10 exercices)

1. Écrire une requête en algèbre relationnelle pour sélectionner tous les étudiants inscrits en 2024.

Étudiant(id\_étudiant, nom, date\_inscription)

2. Extraire les noms des employés qui travaillent dans le département 'Informatique'.

- Employé(id\_emp, nom, code\_dept)
- Département(code\_dept, libelle\_dept)

3. Trouver les clients qui n'ont passé aucune commande.

- Client(id\_client, nom)
- Commande(id\_commande, id\_client)

4. Obtenir la liste des produits dont le prix est supérieur à 100 euros.

Produit(id\_produit, nom, prix)

5. Sélectionner les titres des livres empruntés plus de 5 fois.

- Livre(ISBN, titre)
- Emprunt(ISBN, id\_emprunt)

6. Trouver les étudiants qui suivent tous les cours d'un enseignant donné.

- Inscription(id\_étudiant, id\_cours)
- Enseignement(id\_ens, id\_cours)

7. Lister les noms des fournisseurs qui livrent plus de 3 articles différents.

- Fournisseur(id\_fournisseur, nom)
- LigneLivraison(id\_livraison, id\_article)
- Livraison(id\_livraison, id\_fournisseur)

8. Écrire une opération de jointure naturelle entre les tables Commandes et Clients.

- Commande(id\_commande, id\_client, date\_commande)
- Client(id\_client, nom)

9. Trouver les employés qui n'ont jamais été affectés à un projet.

- Employé(id\_emp, nom)
- Affectation(id\_emp, id\_tâche)
- Projet(id\_pro, libelle\_pro)
- Tâche(id\_tâche, libelle\_tache, id\_pro)

10. Écrire une requête pour obtenir les films projetés dans plus de 2 salles différentes.

Séance(id\_séance, id\_film, id\_salle)