

TD/TP n° 2**Exercice 1. Un hôpital (TD – Conception de tables)**

Le système de gestion hospitalière dispose de certaines informations pour assurer un bon fonctionnement de l'hôpital. Chaque patient est identifié par son nom, prénom, date de naissance, et numéro d'assuré social. À chaque patient est associé l'ensemble des hospitalisations qui le concernent. Pour chaque hospitalisation, on garde les dates (et heure) d'entrée et de sortie, ainsi que le numéro de lit du patient et le médecin responsable. Il peut y avoir d'autres médecins auxiliaires qui ont eu un rôle lors du suivi du patient. De plus on veut pouvoir indiquer qu'une hospitalisation fait suite à une hospitalisation précédente.

1. Concevoir à l'aide d'un diagramme la structure de la base de données.
2. Modifier ce diagramme pour se préparer à l'implémentation en SQL. On pensera en particulier aux types et au cas des cardinalités multiples.

Exercice 2. Une bibliothèque (TD – Conception de tables)

On souhaite créer une base de données qui permette de gérer une bibliothèque.

Dans notre bibliothèque se trouvent des livres qui ont un titre, un auteur, et une année de publication. Avec un titre et un auteur, j'identifie un seul ouvrage. Par contre, il est possible d'avoir plusieurs exemplaires d'un même ouvrage : deux exemplaires auront un code barre différent. Cela est bien pratique, car on souhaite également pouvoir gérer l'emprunt d'un livre par un adhérent : c'est l'exemplaire qui est emprunté, et non l'ouvrage !

Les adhérents ont un numéro d'adhérent ; on garde leur nom et prénom. On veut garder une trace de tous les emprunts passés et en cours : un emprunt spécifie l'adhérent et l'ouvrage concernés, ainsi que la date d'emprunt et éventuellement la date de retour.

1. Concevoir à l'aide d'un diagramme la structure de la base de données.
2. Modifier ce diagramme pour se préparer à l'implémentation en SQL. On pensera en particulier aux types et au clefs primaires.

Exercice 3. Les données de la FFR (TD – Conception de tables)

La fédération française de rugby (FFR) utilise des bases de données pour stocker les informations relatives aux compétitions. Chaque club affilié est identifié par son nom, l'adresse de correspondance et le nom du président. Un club a un ou plusieurs stades, qui ont un nom et une adresse.

Chaque match a lieu entre deux clubs, à une date donnée, dans un stade et est arbitré par un ou plusieurs arbitres (chaque arbitre a un nom, un prénom, une date de naissance et un numéro de licence). On garde également pour chaque match et pour chaque équipe le nombre d'essais, transformations, pénalités, drops, cartons jaunes et cartons rouges.

1. Concevoir à l'aide d'un diagramme la structure de la base de données.
2. Modifier ce diagramme pour se préparer à l'implémentation en SQL.

Exercice 4. La création de mocktails (TD – Conception de tables)

On veut créer un site internet collaboratif de création de mocktails (des cocktails sans alcool). On désire pour cela avoir une base de données d'ingrédients que l'on peut associer dans une recette de mocktail ; on conserve alors la quantité de l'ingrédient que l'on met dans le mocktail. Un mocktail a un nom et est créé par un utilisateur du site, identifié par son pseudo et son adresse. Un utilisateur peut également laisser un commentaire relativement à un mocktail ; un commentaire peut également être marqué comme une réponse à un commentaire précédent.

1. Concevoir à l'aide d'un diagramme la structure de la base de données.
2. Modifier ce diagramme pour se préparer à l'implémentation en SQL.

Exercice 5. Les vols (TD – Conception de tables)

On souhaite répertorier dans une base de donnée les différents vols proposés par des compagnies aériennes. Chaque compagnie a un nom et un code qui l'identifie (par exemple AF pour Air France) et propose plusieurs vols. Un vol a un aéroport de départ et un aéroport d'arrivée, ainsi qu'une date et heure de départ et d'arrivée. On souhaite aussi savoir si le vol est ouvert à la réservation, ainsi que le nombre de places.

Chaque aéroport a un code et un nom (par exemple CDG pour *Paris Charles de Gaulle* et ORY pour *Orly*). On veut garder pour chaque aéroport les heures autorisées de décollage et d'atterrissage.

Les passagers peuvent réserver des billets; ils doivent alors fournir leur nom, prénom, et numéro de passeport. On enregistre pour chaque billet le prix payé.

1. Concevoir à l'aide d'un diagramme la structure de la base de données.
2. Modifier ce diagramme pour se préparer à l'implémentation en SQL.

Exercice 6. Un emploi du temps (TD – Conception de tables)

On souhaite stocker les informations relatives à un emploi du temps. Un créneau est relatif à un module, une salle de classe, un enseignant, et un groupe. On garde bien sûr la date du cours ainsi que les heures de début et de fin.

Le module a un code et un titre et un responsable (un enseignant); le module pointe également vers un diplôme (lui-même ayant un code et un titre!).

Chaque salle de classe a un numéro unique constitué de lettres et de chiffres pour le bâtiment et uniquement de chiffres pour la salle elle-même. On précise que le bâtiment et la salle doivent être stockés séparément afin de pouvoir aisément faire des recherches par bâtiment. Une salle de classe a une capacité donnée, et on garde également une description (par exemple « Salle machine équipée sous Linux uniquement. »).

Un enseignant a un nom et un prénom, ainsi qu'une composante de rattachement (par exemple « IUT de Créteil-Vitry »). Chaque composante a un nom et on souhaite lui associer l'ensemble des diplômes qu'elle délivre. On précise qu'un même diplôme peut être délivré par plusieurs composantes.

1. Concevoir à l'aide d'un diagramme la structure de la base de données.
2. Modifier ce diagramme pour se préparer à l'implémentation en SQL.

Exercice 7. Création de tables en SQL (TP)

1. Se connecter à PostgreSQL comme dans le TP précédent.
2. Pour chaque exercice fait en TD :
 - a) Créer un nouveau schéma.
 - b) Se placer dans ce schéma.
 - c) Créer les tables nécessaires.
 - d) Créer les références entre les tables.
 - e) Insérer quelques tuples dans chaque table.
 - f) Sortir du schéma (en revenant dans le schéma `public`).
3. Pour chaque exercice non fait en TD :
 - a) Le faire.
 - b) Faire la question précédente.
4. Rendre votre production de TP comme dans le TP précédent.