

Module 4 - Exercices : Modélisation, associations

version: 1.0

Exercices sur la modélisation conceptuelle, le [passage du modèle conceptuel au modèle relationnel](#).

Exercice 1 : le SI du mondial de football

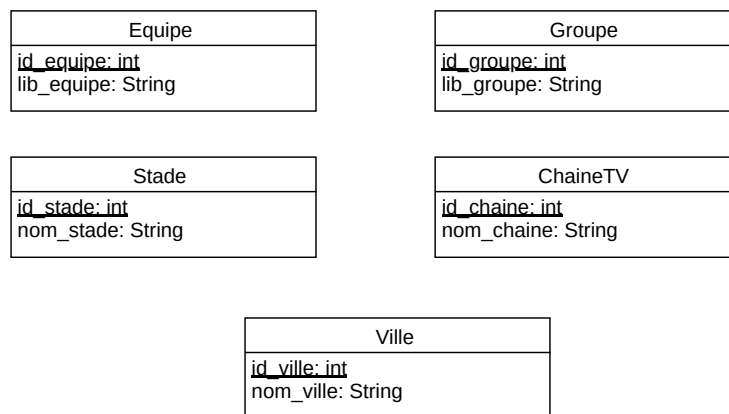
L'objectif de cet exercice est de produire un modèle conceptuel de l'organisation du 1er tour de la coupe du monde de football.

Au premier tour, les équipes sont réparties en 6 groupes de 4 (soit 24 équipes au total à chaque édition). Dans chaque groupe, chaque équipe joue contre les autres, ce qui fait 6 matchs à organiser par groupe. À la fin du 1er tour, seules les deux premières équipes passent au second tour pour continuer la compétition, les autres sont éliminées.

Attribut	Désignation	Type
lib_equipe	Libellé de l'équipe participante (pays)	Chaîne de caractères
lib_groupe	Libellé du groupe (par exemple, "Groupe A")	Chaîne de caractères
nom_ville	Nom de la ville où se déroule le match (par	

	exemple, “Johannesburg”)	Chaîne de caractères
nom_stade	Nom du stade où se déroule le match (par exemple, “Roazhon Park”)	Chaîne de caractères
jour_match	Jour d’un match (par exemple, “11/06/2020”)	Date
heure_match	Heure d’un match (par exemple, “13h30”)	Décimal
score_match	Score d’un match (par exemple “1-2”)	Chaîne de caractères
nom_chaine_tv	Nom de la chaîne TV qui diffuse le match (par exemple, “France 2”)	Chaîne de caractères
classement	Classement de l’équipe au sein du groupe	Entier court

1. **Compléter** le diagramme de classes UML suivant (ajouter les **attributs** manquants et la ou les **classes** manquantes) en vous servant du *dictionnaire des données* qui vous est fourni pour vous aider à démarrer votre conception.



2. Une fois votre diagramme de classes complété, **relier** les classes par des **associations** *judicieusement choisies*. Faites figurer sur chaque association un **nom** et la **cardinalité** à chaque extrémité de l’association.

Source: cet exercice est adapté de l’exercice La déroute des bleus (Modélisation des bases de données: UML et les modèles entité-association, 4e édition, Christian Soutou, 2017, Eyrolles, Exercice 1.1 p 124)

Exercice 2: du modèle conceptuel au modèle relationnel (au modèle physique)

Afin de développer un SI informatisé pour une entreprise de services, nous avons identifié les relations suivantes:

- Client: Nom, Prénom
- Facture: Numéro de facture, Date d'émission de la facture, Montant de la facture en euro, Facture payée ou non

1. À partir de ce dictionnaire des données minimal, **développer** le modèle conceptuel (diagramme de classes UML). **Ajouter** les associations entre classes.
2. **Passer** du modèle conceptuel au modèle relationnel (des classes aux relations/tables). *Penser à ajouter les attributs manquants propres au modèle relationnel.*
3. **Écrire** les requêtes SQL dans un script `ddl_modelisation.sql` et exécutez les via le programme client `mysql` en *batch mode*. Pour cela, créer la base de données `modélisation`.
4. **Insérer** les données suivantes:

- John Doe a deux factures
 - F-900-08, 12/12/2022, 120.5 EUROS, non payé
 - F-500-02, 13/01/2023, 90 EUROS, payé
- Jane Doe a 4 factures
 - Z-500-03, 01/01/23, 1000 EUROS, non payé
 - J-400-02, 11/09/23, 800 EUROS, payé
 - F-434-04, 23/12/22, 400 EUROS, non payé
 - J-333-05, 27/01/23, 1250 EUROS, non payé

5. En utilisant les mots SQL de la DML (Data Manipulation Language: `SELECT`, `WHERE` et opérateurs logiques) écrivez un script `dml_modelisation.sql` afin de récupérer les informations suivantes:
 1. **Lister** toutes les factures impayées
 2. **Lister** toutes les factures impayées dont le montant est supérieur à 500 EUROS
 3. **Lister** toutes les factures de 2022
 4. **Lister** toutes les factures impayées de 2023

Vérifier la cohérence de la base de données (facture bien associée au client correspondant)

Exercice 3 : manipulation des données temporelles

Manipuler les dates et les fonctions temporelles de MySQL

Écrire des requêtes, à l'aide des fonctions temporelles de MySQL, donnant les informations suivantes:

1. La date dans 31 jours
2. La date (date+temps) à 11 heures, 1 microseconde de la date du jour avec 1 jour, 1 microseconde en plus.
3. La date dans 4 mois
4. La date dans 7 jours et 1h30
5. Aujourd'hui, mais en anglais

Exercice 4 : les fonctions, regroupements et opérateurs ensemblistes

Reprendre la base de données **parc** créée au module 3.

Requêtes monotables

Maîtriser ORDER BY, WHERE, AND et OR

Écrire les requêtes pour récupérer les données suivantes:

1. les postes du type 'p8'.
2. Noms des logiciels de type 'UNIX'.
3. Noms, adresses IP, numéros de salle des postes de type 'UNIX' ou 'PCWS'.
4. Même requête pour les postes du segment '130.120.80' triés par numéros de salles décroissants.
5. Numéros des logiciels installés sur le poste 'p6'.
6. Numéros des postes qui hébergent le logiciel 'log1'.
7. Noms et adresses IP complètes (ex : '130.120.80.01') des postes de type 'TX' (utiliser la fonction de concaténation)

Fonctions de regroupement

Comprendre et savoir utiliser les fonctions SUM, AVERAGE, COUNT, HAVING, GROUP BY, ORDER BY, GROUP_CONCAT.

Écrire les requêtes pour récupérer les données suivantes:

8. Pour chaque poste, le nombre de logiciels installés (en utilisant la table Installer).
9. Pour chaque salle, le nombre de postes (à partir de la table Poste).
10. Pour chaque logiciel, le nombre d'installations sur des postes différents.
11. Moyenne des prix des logiciels 'UNIX'.
12. Plus récente date d'achat d'un logiciel.
13. Numéros des postes hébergeant 2 logiciels.
14. Nombre de postes hébergeant 2 logiciels (utiliser la requête précédente en faisant un SELECT dans la clause FROM).

Opérateurs ensemblistes (algèbre relationnelle)

Comprendre et savoir utiliser les opérateurs DISTINCT, UNION, NOT IN

Écrire les requêtes pour récupérer les données suivantes:

15. Types de postes non recensés dans le parc informatique (utiliser la table Types).
16. Types existant à la fois comme types de postes et de logiciels.
17. Types de postes de travail n'étant pas des types de logiciels.

Exercice 5

Modélisation (Niveau conceptuel)

Modéliser le SI d'une station-service, qui dispose de plusieurs pompes. **Proposer** une modélisation conceptuelle, puis **transformer** en modélisation relationnelle (avec les clefs, les types et les contraintes). Suite au contexte actuel, l'état a imposé des restrictions sur le plein faisable en station par personne et par jour. **Ajouter** une contrainte de restriction sur un prix maximum de 120 EUROS lors d'une transaction.

Exercice 6

Modélisation (Niveau conceptuel)

Modéliser le SI d'un cinéma. Le cinéma dispose de 3 salles, et diffuse 5 films différents par mois. Chaque salle propose 3 séances: matin, midi et soir. **Ajouter** une réduction des tarifs en fonction: - de l'âge du client - en dessous de 12 ans: -50% - entre 12 et 16: -25% uniquement le week-end (à partir de la séance de vendredi soir jusqu'à celle du dimanche soir) - de son statut (chômage: -50%)

La réduction doit être appliquée lors de la recherche du prix (**SELECT**) pour une séance.

Exercices et ressources supplémentaires

- [Challenges SQL sur codewars](#), nécessite de se créer un compte
- [Exercices en ligne \(sections 2, 3, 5 et 6\)](#), section 2-Calculs et fonctions, 3-Agrégats, 5 sous requêtes, 6 opérations ensemblistes
- [Les meilleurs cours et tutoriels pour apprendre les bases de données MySQL](#), sélection des meilleurs tutoriels et cours de formation gratuits pour apprendre le Système de Gestion de Base de Données MySQL. Vous trouverez les meilleures méthodes éducatives pour une formation agréable et complète, ainsi que des exercices intéressants, voire ludiques. Une sélection proposée par le site [développez.com](#), un très bon site auquel contribuent des personnes très compétentes en base de données relationnelles

Annexe A: bien débuter un travail de conception

Quelques rappels et conseils sur la démarche à adopter pour débuter un travail de conception:

- établir la liste des attributs sous la forme d'un [dictionnaire des données](#). Document indispensable pour lever les ambiguïtés de langage, communiquer avec votre client ou les experts du métier sur lequel vous travaillez. Chaque attribut doit avoir un nom, une définition, un type, un commentaire (contraintes métiers)
- épurer, éviter les synonymes et les polysémies
- décomposer les attributs structurés (**atomicité**): adresse, numéro de sécurité sociale, etc.
- repérer les identifiants (ce qui identifie une structure, en général un label), en créer si nécessaire
- déduire les classes (entités) à partir de chaque identifiant
- disposer chaque attribut dans une classe (dépendance)
- déterminer les associations entre classes dans un second temps
- itérer
- ne pas raisonner en termes de traitements, mais en termes de *données factuelles et tangibles (les faits, toujours les faits!)*

Références utiles

- [Fixed-Point Types \(Exact Value\) - DECIMAL, NUMERIC](#)

- [Date and Time Functions](#)