

TD : Les contraintes d'intégrité

Exercice 1 :

Une société responsable de l'organisation de manifestations sportives veut se doter d'une base de données relationnelle contenant les informations nécessaires à la gestion des rallyes de véhicules.

Chaque rallye est identifié par son nom (NOMR) et son année d'organisation (AR), a un budget et se caractérise par ses sponsors. Un rallye est divisé en étapes. Chaque étape d'un rallye a un numéro chronologique (NUMET), une ville de départ (VilleD) et une ville d'arrivée (VilleA), ainsi que le nombre de kilomètres à parcourir entre ces deux villes. Une étape peut traverser plusieurs pays (PAYS) et un pays peut être traversé par plusieurs étapes. Pour chaque participant au rallye, on connaît son nom, son prénom, son adresse, son nombre de participations aux rallyes ainsi que sa nationalité. Chaque participant participe au rallye au sein d'un seul équipage. Un équipage est composé de deux participants (pilote et co-pilote).

Les organisateurs attribuent un numéro à chaque équipage (NumEq) pour les différencier tout au long du rallye. Ils font de même pour les véhicules (NumV). Un véhicule doit correspondre à un équipage. Il doit être décrit par sa marque, son modèle et son numéro de série. Il doit appartenir à un groupe qui décrit son niveau de préparation. Il y a 3 groupes possibles : GroupeA, GroupeN et GroupeR. Un modèle de véhicule correspond à une seule marque. Les résultats de chaque étape et pour chaque équipage sont classés. On suppose que le gagnant est celui qui réalise à la fin le minimum de temps.

Le schéma relationnel de la base de données est le suivant :

Rallye (NOMR, AR, budget)

RallyeSponsor (NOMR, AR, sponsor)

EtapeRallye (NOMR, AR, NUMET, VilleD, VilleA, Nbk)

EtapeRallyePays (NOMR, AR, NUMET, PAYS)

Participant (NumP, nom, prénom, adresse, nbp, nationalité)

Equipage (NOMR, AR, NumEq, NumPilote, NumCo-pilote)

Constructeur (Modèle, Marque)

Véhicule (NOMR, AR, NumV, Modèle, nsérie, NumEq, groupe)

Résultats (NOMR, AR, NUMET, NumEq, temps)

1. Déterminer la clé primaire et éventuellement les clés étrangères de chaque relation.
2. Ecrire la commande SQL de création de la table EtapeRallye en précisant toutes les contraintes d'intégrité possibles.
En supposant que toutes les tables ont été créées, A partir de l'énoncé, déterminer :
 3. une contrainte de valeur et écrire la commande permettant de l'ajouter (on nommera cette contrainte cont_val).
 4. une contrainte intra-relation et écrire la commande permettant de l'ajouter.
 5. Soit NBV, le nombre de véhicules participant dans un rallye. Ecrire la commande SQL, permettant d'ajouter l'attribut NBV dans la base de données.
 6. Ecrire la commande SQL permettant d'initialiser NBV.
 7. Quelles sont les répercussions de ces commandes (questions 6 et 7) sur les catalogues
 8. Soit CONTRAINTE le nom du catalogue système permettant de gérer les contraintes d'intégrité statiques.
 - a) Proposer une structure pour ce catalogue (expliquer les différents attributs de ce catalogue)
 - b) Donner les répercussions sur les catalogues de l'exécution de la commande de la question 4.

Exercice 2 :

Soit la base de données de la Gestion des spectacles d'un cirque, ayant le schéma relationnel suivant :

PROGRAMME (DATE, HEURE, MINUTE, NUMS)

EMPLOYE(NUMS, NOME, DATEN, ANCIENNETE)

SPECTACLE (NUMS, TITRE, THEME, RESPONSABLE)

PARTICIPER (JOUEUR, NUMS, ROLE)

ANIMAL (NOMANIMAL, ESPECE, DRESSEUR)

NUMERO_ANIMAL (NOMANIMAL, NUMS)

Tickets (NUM-TICKET, DATE, NBP, ZONE)

TARIF (ZONE, PRIX_PLACE, NBPLACES, NBPLAC_DISPO)

Le cirque donne plusieurs spectacles par jour. Le même spectacle peut se répéter plusieurs fois dans la semaine. Chaque spectacle identifié par un numéro, a un titre, un thème et un responsable. Le thème peut prendre comme valeur: otaries, jeux d'eau, jonglage, équilibre, clown, grands classiques, sensations. Un employé du cirque peut participer à des spectacles (joueur), dresser des animaux (dresseur) ou être responsable de spectacles. Un joueur participant à un spectacle a un rôle pouvant être: clown, acrobate, funambule, jongleur. Le cirque possède des animaux de différentes espèces: lion, tigre, hippopotame, crocodile, zèbre, serpent, poney, otarie. Chaque animal a un dresseur et peut participer à plusieurs spectacles. Dans un même spectacle, on peut avoir plusieurs joueurs et/ou plusieurs animaux. NUMÉ, JOUEUR, DRESSEUR et RESPONSABLE sont du même domaine.

L'espace réservé aux spectateurs est divisé en zones (Par ex. z1: 1ers rangs, z2: milieu, z3: au fond...). Chaque zone a un nombre donné de places (NBPLACES). Le prix d'une place (PRIX_PLACE) varie selon la zone (par ex. 1ers rangs: 1600 DA - milieu : 1300 - au fond : 900 ...). NBPLAC_DISPO : Nombre de places disponibles dans une zone. On donne à un client qui réserve, un ticket qui indique la date, la zone et le nombre de places (NBP) réservées. Deux spectacles différents ne peuvent pas avoir le même titre.

On voudrait toujours connaître le titre et le thème d'un spectacle ainsi que le dresseur d'un animal.

NumS, NumE : nombres sur 3 chiffres. Les noms, titre, thème et rôle sont des chaînes fixes de 30 caractères.

1. Déterminer les clés primaires et les clés étrangères des relations.
2. Déterminer les contraintes d'intégrité statiques de cette BD.

Type de contrainte	Relation	La contrainte

3. Ecrire la commande de création de la relation SPECTACLE en précisant toutes les contraintes d'intégrité statiques possibles.
4. Ecrire les commandes permettant d'ajouter les contraintes suivantes :
 - a- sur l'espèce d'un animal.
 - b- une contrainte d'unicité.
 - c- une contrainte d'attribut obligatoire.
5. Quelles sont les répercussions de la commande de la Question n°3 sur les catalogues
6. On voudrait gérer automatiquement la vente des tickets comme suit :
 - Créer un trigger pour décrémenter l'attribut NBPLAC_DISPO suite à la vente d'un ticket.

Exercice 3 :

Un groupe industriel veut gérer de manière centralisée les renseignements concernant ses différentes usines. La base de données construite a le schéma relationnel suivant :

Usine (Nom-U , Adresse-U , Téléphone-U , Responsable-U)
Employé (Num-E , Nom-E , Prénom-E , Adresse-E)
Point-Vente (Num-P , Nom-P , Adresse-P , Nom-U)
Produit (CodeP , DésignationP , Prix_U_P)
Client (Num-Cl , Nom-Cl , Prénom-Cl, Adresse-Cl)
Employ_PointV (Num-E , Date-Af , Num-P)
Stock (Num-P, CodeP , QtéStk, QtéVdu)
Vente_Client (Num-P, NumCl, CodeP, Qté)

Chaque usine fabrique des produits et est constituée d'un certain nombre de points de vente pour les vendre. Un point de vente correspond à une seule usine. Un point de vente peut vendre plusieurs produits et un même produit peut être vendu dans différents points de vente.

Un employé peut être affecté à plusieurs points de vente dans le temps. Mais à une date donnée, il n'est affecté qu'à un seul point de vente

Un client peut acheter différents produits dans un ou plusieurs points de vente

Responsable-U : numéro de l'employé qui est responsable de l'usine

Date-Af : date d'affectation d'un employé à un point de vente

QtéStk : quantité en stock d'un produit dans un point de vente

QtéVdu : cumul (total) des quantités vendues d'un produit dans un point de vente

Qté : quantité d'un produit vendu à un client au niveau d'un point de vente.

Les adresses sont définies sur des chaînes de 50 caractères maximum
 Les noms sont définis sur des chaînes de 30 caractères maximum
 Le responsable d'une usine est un employé. Il ne peut être responsable que d'une usine à la fois.
 On voudrait toujours connaître les responsables des usines, les noms des employés, les noms des points de vente et les désignations des produits.
 Le prix unitaire d'un produit est compris entre 100 et 1500 DA

1. Déterminer les clés primaires et les clés étrangères.
2. Ecrire la commande SQL de création de la table Point-Vente en précisant toutes les contraintes d'intégrité possibles.
3. A partir de l'énoncé, déterminer une contrainte d'unicité et écrire la commande permettant de l'ajouter.
4. Ecrire la commande SQL permettant d'ajouter une contrainte de valeur (déterminée à partir de l'énoncé)
5. Le point de vente N° 30 vend au client N°340 une quantité 100 du produit N° 210. Ecrire la commande SQL qui permet d'enregistrer cette information
6. Quelles sont les répercussions de cette commande sur les catalogues.
7. Toute opération de vente comme celle de la Question 5 nous oblige à modifier les quantités des produits pour assurer la cohérence de la BD. Nous voudrions que les quantités en stock et vendues des produits soient mises à jour automatiquement. Définir une contrainte d'intégrité dynamique permettant de le faire.

Exercice 4 :

On veut gérer de manière centralisée les renseignements concernant les laboratoires de recherche des différentes universités. La base de données construite a le schéma relationnel suivant :

Université (CodeUniversité, NomUniversité)

Laboratoire (CodeLab, NomLab, Chef_Lab, Date-création, CodeUniversité, Siteweb).

Chercheur (NumCh, NomCh, grade, CodeLab)

Prime_Mensuelle (Grade, MontantP) MontantP: montant mensuel (1 mois) de prime correspondant à 1 grade

Projet (CodePrj, Durée, Chef_Prj, Budget) Durée : durée d'un projet (en mois)

Membre (NumCh, CodePrj, Rôle, ChargHoraire)

Les chefs de laboratoires et les chefs de projets sont des chercheurs (les attributs Chef_Lab, Chef_Prj et NumCh sont du même domaine.

1. Ecrire la commande SQL de création de la table Membre en précisant toutes les contraintes d'intégrité possibles. Ci-dessous, la description de la table membre.

Un chercheur peut être membre dans plusieurs projets. Un projet peut être réalisé par plusieurs chercheurs. L'attribut Rôle définit le rôle joué par le chercheur dans un projet et ChargHoraire indique le nombre d'heures effectué par un chercheur dans un projet. Toutes les informations de cette table (Membre) sont obligatoires. Numch nombre de 7 chiffres, codPrj chaîne de 10 caractères et ChargHoraire nombre de 5 chiffres

2. Peut-on définir une contrainte intra-relation horizontale dans la table membre? justifier
3. L'administrateur veut garder le total de la charge horaire de chaque chercheur et chaque laboratoire. Pour cela, il ajoute deux attributs : **TotalHrChercheur** dans Chercheur et **TotalHrLabo** dans Laboratoire.
 - a) Ecrire les commandes permettant d'ajouter ces deux attributs
 - b) Ecrire la commande de création du trigger **TotalHrChercheur_trigger** qui met à jour l'attribut totalHrChercheur après l'insertion d'un membre à un projet.
 - c) Ecrire la commande de création du trigger **TotalHrLabo_trigger** qui met à jour l'attribut totalHrLabo après l'insertion d'un membre à un projet.
 - d) Un chercheur peut changer de laboratoire. Créer un trigger qui met à jour l'attribut TotalHrLabo des deux laboratoires concernés par cette opération.

Exercice 5 :

Nous considérons un système centralisé de réservation de chambres d'hôtels, basé sur le schéma relationnel suivant :

CLIENT (NumCl, NomCl , Tel , Adresse)

HOTEL (NumH, NomH, CatH, AdrH, TelH, NbChDisp)

CHAMBRE (NumCh, NumH , CatCh)

RESERVATION (NumRes, NumH, NumCl, DateDebRes, DateFinRes)

CHAMBRE_RESERVEE (NumRes, NumCh)

Où,

-NumCl, NumH et NumRes sont uniques dans la BD

-CatH : catégorie d'un hôtel décrivant s'il est à 1 étoile, 2 étoiles, ...5 étoiles

-NumCh est un entier indiquant le numéro de la chambre.

- CatCh : catégorie d'une chambre pouvant être soit simple ou double.

- NbChDisp : nombre de chambres disponibles dans un hotel.

-Un client peut réserver plusieurs chambres dans une même réservation

-On voudrait toujours connaître le nom et le téléphone d'un client ainsi que le nom d'un hotel.

1. Déterminer les clés primaires et les clés étrangères des relations.
2. Déterminer les contraintes d'intégrité statiques de cette BD.
3. Ecrire la commande SQL de création de la table Chambre en précisant toutes les contraintes d'intégrité statiques possibles.
4. Ecrire la commande permettant d'ajouter la contrainte sur la catégorie d'hôtel
5. Deux hôtels différents ne peuvent pas avoir le même nom. Ecrire la commande permettant d'ajouter cette contrainte.
6. Définir une contrainte d'intégrité dynamique relative à la réservation d'une chambre d'hôtel.

Exercice 6 :

Soit la base de données suivante :

SATELLITE (NomSat, Position , Débit)

CHAINE (NomChaine, Langue)

SAT-CHAINE (NomSat, NomChaine, Fréquence, Polarité, Mode)

GUIDE-TV (NomChaine, date, HeureDeb, Programme, Catégorie)

1. Ecrire la commande de création de la table GUIDE-TV en précisant toutes les contraintes
Sachant que : NomChaine, Programme et Catégorie sont des chaînes de 50 caractères au max
HeureDeb est définie sur 5 caractères.
La catégorie d'un programme peut être Cinéma, Documentaire, Série, Enfant ou Sport.
2. Ecrire la commande permettant d'ajouter la contrainte spécifiant que l'attribut Mode prend ses valeurs dans le domaine ('Crypté', 'Clair').
3. Ecrire la commande permettant d'ajouter l'information sur le nombre de chaînes abritées dans un satellite.
4. Définir une contraintes d'intégrité dynamique relatives à l'insertion/suppression d'une chaîne dans/d'un satellite.