

 Se former autrement	<h1>EXAMEN</h1>	
	Semestr : 1 <input type="checkbox"/> 2 <input checked="" type="checkbox"/>	
Session : Principale <input checked="" type="checkbox"/> Rattrapage <input type="checkbox"/>		
Module : SGBD Enseignants : Équipe SGBD Classes : 3A Documents autorisés : OUI <input type="checkbox"/> NON <input checked="" type="checkbox"/>		
Date : 17-05-2018		Nombre de pages : 02
Heure : 10h30		Durée : 01h30

Nous souhaitons créer une application qui permet la gestion d'audience de diverses chaînes télévisées. Une partie du schéma relationnel de l'application est donnée ci-dessous :

CHAINETV (idCH, nomCH, typeCH)
 ANIMATEUR (idA, nomA, nationalité, gender)
 PROGRAMME (idP, #idCH, #idA, titre, typeP, périodicité, jour, heure_debut, durée)
 SONDAGE (heureDeb, heureFin, dateS, #(idP, idCH), audimat)

Remarques :

- La **périodicité** peut être (*quotidienne, hebdomadaire*).
- Le type de chaîne, **typeCh**, peut être (*ipTV, TNT, satellite*).
- Le type de programme, **typeP**, peut être (*politique, sport, news, documentaire, divertissement*).
- L'**Audimat** correspond au *nombre* de téléspectateurs qui regardent le programme à un moment donné.
- La **durée** d'un programme est exprimée en *minutes*.
- On suppose qu'un programme n'est diffusé qu'une seule fois par jour.

Travail demandé :

1. Créer une vue **V_PROG_MENS** qui permet d'afficher le titre du programme, le nom de l'animateur ainsi que le nom de la chaîne pour les programmes dont la **périodicité** est *hebdomadaire*. (2 pts)

2. Écrire une procédure stockée **PS_AUD_PROG** qui permet d'afficher le total des audimats pour chaque type de programme. (2 pts)
3. Écrire une fonction stockée **FN_NB_PROG(p_idA number)** qui permet de retourner le nombre de programmes présentés par l'animateur dont l'identifiant est donné en paramètre. Si l'animateur n'existe pas la fonction doit retourner -1. (2 pts)
4. Écrire une procédure stockée **PS_NOM_PROG** qui permet d'afficher pour chaque animateur, ayant plus de deux programmes, son nom et sa nationalité ainsi que les titres, types et périodicité des programmes qu'il présente. Utiliser la fonction **FN_NB_PROG**. (3 pts)
5. Écrire une fonction stockée **FN_POUR_AUDIMAT (p_idCH number, p_date date)** qui retourne, pour une date donnée, le pourcentage de l'audimat (voir encadré ci-dessous) d'une chaîne TV dont l'identifiant est donné en paramètre. Si la chaîne TV n'existe pas la fonction doit retourner -1. (3 pts)

Le pourcentage de l'audimat d'une chaîne = $\text{audimat total chaîne} / \text{audimat total}$

6. Écrire une procédure stockée **PS_CLASS_CHAINE (p_date date)** qui affiche, pour une date donnée, une liste numérotée des noms des chaînes selon un ordre décroissant du pourcentage de leur audimat. Utiliser la fonction **FN_POUR_AUDIMAT**. (2 pts)
7. Écrire une procédure stockée **PS_TOP_PROG (p_heureDeb number, p_heureFin number, p_date date)** qui permet d'afficher le nom du(des) programme(s) qui a(ont) réalisé l'audimat le plus élevé pour la plage horaire et la date indiquées en paramètre. Traiter les exceptions suivantes :
 - L'heure de fin doit être supérieure à l'heure de début
 - La date ne doit pas être supérieure à la date d'hier. (3 pts)
8. Écrire un trigger **TRIG_PERIODICITE** qui permet de vérifier la valeur de la colonne *jour* lors de l'insertion au niveau de la table **PROGRAMME** :
 - a. si la périodicité est '**quotidienne**' la colonne *jour* sera nulle.
 - b. si la périodicité est '**hebdomadaire**', la colonne *jour* doit avoir une valeur, sinon, l'insertion doit être bloquée. (3 pts)

Bon courage