Licence Science et Technologies
Devoir surveillé - 12 mars 2015

Université Joseph Fourier GRENOBLE UFR IM²AG

Durée : 2 heures **Le sujet comporte 2 questions indépendantes**
Document autorisé : feuille A4, recto-verso, manuscripte
Le barème est donné à titre indicatif

1 A propos de transport

On reprend le sujet traité en TD/TP sur la base de données "Transports".

1.1 Schéma de relations

Le schéma de la base de données est rappelé ci-dessous (les identifiants sont soulignés) :

LesPersonnes (noP, nom, prénom, salaire)
/ <p, no, pr, s> ∈ LesPersonnes ⟺ la personne identifiée par p a pour nom no, et prénom pr. Elle perçoit un salaire s. */*
LesClients (noC, nom, adresse)
/ <c, n, a> ∈ LesClients ⟺ le client identifié par c a pour nom n et réside à l'adresse a. */*
LesCamions (immat, marque, dateAchat)
/ <i, m, d> ∈ LesCamions ⟺ le camion d'immatriculation i est de marque m. Il a été acheté par l'entreprise à la date d. */*
LesContrats (noTr, dateDep, dateArr, vDep, vArr, noC, immat, noP)
/ <t, dd, da, vd, va, c, i, p> ∈ LesContrats ⟺ le contrat identifié par t est établi pour le compte du client c. Il s'agit du transport depuis la ville vd vers la ville va. Les dates de départ et d'arrivée prévues sont respectivement da et dd. Le camion affecté à ce contrat est celui d'immatriculation i, conduit par le chauffeur p. */*
LesSecrétaires (noP, noC)
/ <p, c> ∈ LesSecrétaires ⟺ la personne secrétaire d'identifiant p est responsable des contrats du client d'identifiant c. */*
LesChauffeurs (noP, nbK)
/ <p, k> ∈ LesChauffeurs ⟺ il est prévu que le chauffeur identifié par p effectue k kms pour le compte de l'entreprise. k vaut 0 pour les chauffeurs affectés à aucun contrat. */*
LesDistances (vDép, vArr, nbK)
/ <d, a, k> ∈ LesDistances ⟺ k est la distance en km entre la ville d et la ville a. */*
Les domaines sont :
domaine (noP) = domaine (noTr) = domaine (noC) = entiers > 0
domaine (nom) = domaine (prénom) = domaine (immat) = domaine (marque) = chaînes de caractères
domaine (dateDep) = domaine (dateArr) = domaine (dateAchat) = date (à l'unité Jour)
domaine (adresse) = domaine (vDep) = domaine (vArr) = {"Grenoble", "Paris", "Pau", etc.}
domaine (nbK) = entiers ≥ 0, domaine (salaire) = réels > 0

Les contraintes d'intégrité :

LesPersonnes[noP] = LesSecrétaires[noP] ∪ LesChauffeurs[noP]
LesSecrétaires[noP] ∩ LesChauffeurs[noP] = ∅
LesSecrétaires[noC] ⊆ LesClients[noC] LesContrats[noC] ⊆ LesClients[noC]
LesContrats[immat] ⊆ LesCamions[immat] LesContrats[noP] ⊆ LesChauffeurs[noP]
LesContrats[vdep, varr] ⊆ LesDistances[vdep, varr]

D'autres contraintes :

1. La relation LesDistances est symétrique, c'est-à-dire :
   <v1, v2> ∈ LesDistances ⟺ <v2, v1> ∈ LesDistances.

2. La relation LesDistances n'est pas réflexive : <v1, v2> ∈ LesDistances ⟹ v1 ≠ v2

3. Pour chaque chauffeur, le nombre de kilomètres qui lui est associé dans la relation LesChauffeurs est égal à la somme des distances effectuées sur tous ses contrats (passés et prévus). Ce nombre est 0 pour les chauffeurs affectés à aucun contrat.

4. La date de départ d'un transport est inférieure ou égale à la date d'arrivée.

5. La date d'achat d'un camion est inférieure ou égale aux dates de départ des transports auxquels il est affecté.

6. Chaque jour un camion est affecté à au plus un contrat.

7. Chaque jour un chauffeur est affecté à au plus un contrat.

1.2 Implantation de la base de données

On donne le code SQL-Oracle permettant de créer les relations LesCamions, LesPersonnes et LesClients :

```
create table lescamions (
   immat varchar2 (20), marque varchar2 (20), dateAchat date,
   constraint cam_pk primary key (immat) );

create table lespersonnes (
   nop number(4),  nom varchar2 (20), salaire number (6,2),
   constraint pers_pk primary key (nop),
   constraint pers_ck1 check (nop > 0),
   constraint pers_ck2 check (salaire > 0) );

create table lesclients (
   noc number(4), nom varchar2 (20), adresse varchar2 (20),
   constraint clie_pk primary key (noC),
   constraint clie_ck check (noc > 0) );
```

Question 1 (6 points) :
Donner en SQL le code des relations LesDistances, LesSecrétaires, LesContrats et LesChauffeurs.

Question 2 (4 points) :
Quelles sont les contraintes qui restent à la charge des applications (c'est-à-dire celles que l'on ne peut pas exprimer en SQL dans le code de création des relations) ?

1.3 Evaluation de requêtes SQL

Question 3 (5 points) :
En considérant les relations fournies en annexe, donner le résultat retourné par chacune des requêtes ci-dessous (considérer le système Oracle vu en TP). Il peut s'agir d'une erreur, ou de n-uplets affichés sous forme de tableau.

1. select noP, count(*) from LesSecretaires group by noP;

2. select notr, vdep, varr, nbk, noC
   from LesContrats natural join LesDistances
   where nbK $<$ 350;

3. ```
   select noTr, to_char(dateDep, 'Day, DD-Mon-YYYY') as JourDep,
                to_char(dateArr, 'Day, DD-Mon-YYYY') as JourArr,
                (dateArr - dateDep) as duree
   from LesContrats
   where (dateArr - dateDep) in (select max(dateArr - dateDep) from LesContrats);
   ```

   Indication : le 1er janvier 2007 était un lundi.

4. ```
   select noP, sum(di.nbK) as nkkTot, ch.nbK
   from LesChauffeurs ch natural join LesContrats
        join LesDistances di using (vDep, vArr)
   group by noP, ch.nbK
   ```

1.4 Expression de requêtes SQL

Question 4 (5 points) :
Exprimer en SQL les requêtes ci-dessous en respectant rigoureusement les contraintes données.

• Les requêtes devront construire des résultats sans répétition de valeurs, la clause distinct ne sera utilisée que lorsque nécessaire.

• Les produits de relation seront exprimés dans la clause from des requêtes.

• Toute requête emboîtée dans une clause from devra être spécifiée avec soin, de même pour les vues.

1. Donner le numéro, le nom et le prénom des chauffeurs qui ont conduit au moins 2 camions différents?

   Indications pour la requête 1 : la requête ne doit pas contenir les opérateurs IN ou NOT IN, ni GROUP BY et HAVING.

2. Donner le numéro, le nom et le prénom des chauffeurs qui ont conduit exactement un camion.

   Indications pour la requête 2 : la requête ne doit pas contenir les opérateurs IN ou NOT IN, ni GROUP BY et HAVING.

3. Donner le numéro, le nom et le prénom des chauffeurs qui ont conduit tous les camions.

4. Donner le numéro, le nom et le prénom des chauffeurs qui ont été affectés à tous les contrats du client de numéro 4.

5. Pour chaque camion donné par son immatriculation, quel est le chauffeur avec lequel il a effectué (ou va effectuer) le plus de kilomètres.

A Etat de la bases de données Transports

LesDistances :

VDEP	VARR	NBK
Lyon	Marseille	314
Lyon	Paris	466
Lyon	Grenoble	112
Marseille	Grenoble	306
Marseille	Lyon	314
Marseille	Paris	774
Grenoble	Marseille	306
Grenoble	Paris	574
Grenoble	Lyon	112
Paris	Marseille	774
Paris	Lyon	466
Paris	Grenoble	574

LesPersonnes :

NOP	NOM	PRENOM	SALAIRE
1	Peyrin	Jean-Pierre	68
9	Bouchenak	Sara	140
8	Fauvet	Marie-Christine	140
10	Frehse	Goran	130
2	Fernandez	Jean-Claude	160
3	Maltese	Corto	160
11	L'enchanteur	Merlin	160
12	Tata	Catherine	160
13	Tonton	Michael	160

LesClients :

NOC	NOM	ADRESSE
1	Mickey Mouse	Echirolles
2	Tintin	Bruxelles
3	Rantanplan	Washington
4	Mafalda	Gap

LesChauffeurs :

NOP	NBK
9	620
1	154999
10	1706
11	466
12	0

LesCamions :

IMMAT	MARQUE	DATEACHAT
18 CHX 38	renault	14-JUL-07
19 CHX 38	renault	14-JUL-07
20 CHX 38	renault	14-JUL-07
1675 WZ 38	renault	01-MAY-94
1 A 38	renault	01-JAN-45
181 CHX 38	daf	17-JUL-07

LesSecrétaires:

NOP	8	8	13
NOC	1	2	3

LesContrats :

NOTR	DATEDEP	DATEARR	VDEP	VARR	NOC	IMMAT	NOP
1	14-JUL-07	21-JUL-07	Lyon	Marseille	1	18 CHX 38	2
2	01-JAN-07	21-FEB-07	Lyon	Marseille	1	19 CHX 38	9
3	01-FEB-07	28-FEB-07	Grenoble	Marseille	2	1675 WZ 38	9
4	01-FEB-07	28-FEB-07	Paris	Marseille	2	1675 WZ 38	10
7	01-FEB-07	20-FEB-07	Lyon	Paris	3	1675 WZ 38	10
5	01-AUG-06	07-AUG-06	Lyon	Paris	1	1675 WZ 38	10
8	05-AUG-06	17-AUG-06	Lyon	Paris	2	1675 WZ 38	11
9	15-SEP-06	17-SEP-06	Lyon	Paris	4	18 CHX 38	11
108	25-SEP-06	27-SEP-06	Lyon	Paris	4	19 CHX 38	11
118	05-OCT-06	17-OCT-06	Lyon	Paris	4	20 CHX 38	11
10	05-NOV-06	17-NOV-06	Lyon	Paris	4	18 CHX 38	11
11	05-NOV-08	17-NOV-08	Lyon	Paris	4	181 CHX 38	11
12	05-DEC-08	01-JAN-09	Lyon	Paris	4	1 A 38	11
119	15-SEP-06	17-SEP-06	Lyon	Grenoble	2	18 CHX 38	9
138	25-SEP-06	27-SEP-06	Lyon	Paris	1	19 CHX 38	9
140	05-OCT-06	17-OCT-06	Grenoble	Paris	2	20 CHX 38	9
150	05-NOV-06	17-NOV-06	Lyon	Paris	3	18 CHX 38	9
151	05-NOV-08	17-NOV-08	Lyon	Marseille	3	181 CHX 38	9
152	05-DEC-08	01-JAN-09	Marseille	Paris	1	1 A 38	9

TP2 : mise en œuvre de l'application de théâtre

Université Joseph Fourier GRENOBLE UFR IM²AG

B Instructions pour le TP (à réaliser en binôme)

Le compte-rendu du TP (un par binôme) est à déposer sur moodle avant le 10/5/2015 (http://imag-moodle.e.ujf-grenoble.fr) (attention à l'heure de fermeture du site).
Délai de rigueur : tout TP non rendu dans les délais sera noté 0/20.
Le compte-rendu (en 2 fichiers) devra contenir :

- La rédaction des réponses aux questions 5 (le code SQL de création des relations de votre base de données), 6 et 7 de l'énoncé. Ce fichier devra être au format pdf.
- Une archive compressée (par exemple) ".tar.gz" contenant les programmes PHP ont été réalisés. Ces programmes seront testés et évalués par l'enseignant chargé de la correction des TPs.

Tout code fourni doit être commenté.

C Objectifs du TP

Le projet consiste en la réalisation d'une application accessible par internet. L'accent est mis sur le maintien de la cohérence des données. Les technologies utilisées sont :

- PHP et HTML pour la partie applicative et les accès via le web,
- Oracle-SQL pour la partie base de données.

L'objectif est une meilleure compréhension des rôles respectifs de ces technologies lors de leur utilisation dans la perspective de l'implantation d'une application de bases de données.

D Position du problème

Un directeur de théâtre désire informatiser son système de réservation de places pour les spectacles qui se déroulent dans son théâtre au cours d'une même saison (de septembre à juin). A la fin d'une saison, le contenu de la base de données est archivé et vidé. Pour les besoins du projet les données de l'application décrites ci-après ont été simplifiées.

Les spectacles. Un spectacle est identifié par un numéro et on connaît son nom. Un spectacle fait généralement l'objet de plusieurs représentations proposées à des moments différents. Une représentation débute à un moment donné exprimé à la granularité de l'heure (par exemple 28/11/2007 20H).

La salle. La salle est partitionnée en zones numérotées, regroupant chacune un ensemble de places. Une place est identifiée par un numéro de rang, et un numéro de place dans le rang. Une zone est associée à une catégorie tarifaire (orchestre, balcon, poulailler, etc). Toutes les places de la même zone sont dans la même catégorie. Le tarif de base associé à chaque catégorie est fixé pour l'ensemble de tous les spectacles.

Les ventes. Chaque place vendue fait l'objet de l'émission d'un ticket identifié par un numéro de série et estampillé par la date au moment de la transaction (instant à la granularité de la seconde). Un

achat peut concerner plusieurs places. Chaque achat se traduit par la création d'un dossier (identifié par un numéro) auquel est associé le prix global des places.

Hypothèses supplémentaires. Le théâtre conserve toujours 70 places (toutes catégories confondues) qui seront vendues au guichet, dans l'heure qui précède le début du spectacle.

E Modélisation relationnelle

Pour accéder à ces relations avec Oracle, le préfixe `theatre` doit être utilisé. Par exemple : `select * from theatre.LesSpectacles`.

Le schéma relationnel est donné ci-dessous. Les clefs candidates sont formées des attributs notés en caractères soulignés :

LesZones (numZ, nomC)
/ <z, c> ∈ LesZones ⟺ les places de la zone z sont dans la catégorie c. */*
LesCatégories (nomC, prix)
/ <c, p> ∈ LesCatégories ⟺ p est le prix des places se situant dans une zone de catégorie c. */*
LesPlaces (noPlace, noRang, numZ)
/ <p, r, z> ∈ LesPlaces ⟺ la place de numéro p dans le rang r est dans la zone z. */*
LesSpectacles (numS, nomS)
/ <s, t> ∈ LesSpectacles ⟺ le spectacle de numéro s a pour titre t. */*
LesReprésentations (dateRep, numS)
/ <d, s> ∈ LesReprésentations ⟺ d est la date d'une représentation pour le spectacle s. */*
LesTickets (noSerie, numS, dateRep, noPlace, noRang, dateEmission, noDossier)
/ <t, s, d, p, r, e, n> ∈ LesTickets ⟺ le ticket dont le numéro de série est t correspondant la place <p, r> pour la représentation <s, d>, a été émis à la date e, il est associé au dossier n. e < d. */*
LesDossiers (noDossier, montant)
/ <d, m> ∈ LesDossiers ⟺ m est le montant total des places associées au dossier d. */*

domaine (dateRep) = date(heure) */* par ex. 24/11/2007 20H */*
domaine (dateEmission)= date (seconde) */* par ex. 24/11/2007 9H30MN14S */*
domaine (catégorie) = {'orchestre', 'balcon', 'poulailler'}
domaine (numZ) = domaine (noPlace) = domaine (noRang) = domaine (numS) = domaine (noDossier)
= entier > 0
domaine (prix) = réels > 0

Les autres contraintes d'intégrité :

• Pour chaque représentation, il y a 70 places disponibles jusqu'à t - 1 heure (t est l'instant de début de la représentation).
• Dans la relation LesTickets : dateEmission < dateRep
• LesZones[nomC] ⊂ LesCatégories[nomC]
• LesPlaces[numZ] ⊂ LesZones[numZ]
• LesReprésentations[numS] = LesSpectacles[numS]
• LesTickets[numS, dateRep] ⊂ LesReprésentations[numS, dateRep]
• LesTickets[noPlace, noRang] ⊂ LesPlaces[noPlace, noRang]
• LesTickets[noDossier] ⊂ LesDossiers[noDossier]

F Test de l'application fournie

Le serveur `goedel.e.ujf-grenoble.fr` donne accès aux programmes PHP.

Par exemple, l'accès à `http://goedel.e.ujf-grenoble.fr/~vigouroc` (avec n'importe quel navigateur comme `firefox`) exécutera le script `index.php` présent dans le répertoire `public_html` de l'utilisateur `vigouroc`. Si ce répertoire ne contient aucun programme PHP appelé `index.php`, le contenu du répertoire est alors affiché.

Question 5 :
Tester l'application disponible à l'adresse :
`http://goedel.e.ujf-grenoble.fr/~vigouroc/TheatreEtudiants`
Se connecter en utilisant le login usuel (sans _ujf) et le mot de passe bd2015.
Pour chaque fonctionnalité, observer son comportement et regarder le programme correspondant.

G Préparation de l'environnement

Télécharger le squelette de l'application à partir de la plate-forme moodle : sauvegarder le fichier `Theatre.tar.gz` dans son répertoire `public_html` personnel.
Taper `gunzip Theatre.tar.gz` puis `tar xvf Theatre.tar` tous les fichiers seront extraits de l'archive.
Voir à l'adresse `http://imag-wiki.e.ujf-grenoble.fr/doku.php?id=faq#service_web_sur_goedel` la procédure pour que le répertoire `public_html` soit accessible depuis un navigateur.
Maintenant, vous pouvez accéder à votre propre application à l'adresse `http://goedel.e.ujf-grenoble.fr/~votre-login`.

H Interroger la base de données fournie

Pour toutes les questions de cette section, les relations à utiliser doivent être préfixées par `theatre.` Exemple : pour utiliser `LesSpectacles`, noter `theatre.LesSpectacles`
Question 6 :
Modifier les programmes `DetailsRepresentation.php` et `DetailsRepresentation-action.php` pour :

1. Afficher la date de la représentation (dans la granularité de l'heure) et le nom du spectacle.
2. Afficher les détails des représentations d'un spectacle choisi dans la liste.

Question 7 :
Modifier les programmes `CategorieTicket.php` et `Categorie-action.php` pour améliorer l'interface utilisateur.

Question 8 :
Sur le modèle des scripts fournis, compléter l'application afin de réaliser les fonctions présentes dans menu (partie *Requêtes à réaliser sur la BD fournie*).

I Créer sa propre base de données

A partir de cette section, les relations utilisées sont celles créées personnellement. Elles ne sont donc plus préfixées par `theatre.` Chacun possède les relations qu'il a créées (accessibles sans préfixe par celui qui les a créées).
Question 9 :
Donner le code SQL (Oracle) permettant de programmer les relations spécifiées dans la section **??**.
Créer sa propre base de données sur la machine `im2ag-oracle.e.ujf-grenoble.fr` et la remplir avec les données de la base de données fournie.

Question 10 :
Identifier les contraintes d'intégrité spécifiées dans la section **??** et qui ne sont pas exprimées en SQL, et qui par conséquent, sont à la charge de l'application.

J Implantation de l'application Web

Question 11 :
Indiquer en quoi diffèrent les programmes des trois versions fournies pour la mise à jour "Une nouvelle représentation du spectacle *Les enfoirés* est programmée le 29/2/2016".

- Quelles sont les différences dans leur comportement ?
- A quels endroits dans le code trouve t-on le traitement de ces erreurs ?

Question 12 :
Compléter l'application (voir les fonctions demandées dans le menu, partie *Requêtes à réaliser sur la BD à créer*).