

L2 INF245 — Examen mai 2015, session 1



Durée : 2H

Documents autorisés : une feuille A4, recto-verso, manuscrite, et le document "Notice PHP (ou le minimum vital)"

1 A propos de relations

Soient R et S deux relations dont les schémas et les valeurs sont donnés dans les tableaux suivants :

R	A	B
	1	11
	2	20
	4	21
	3	10

S	B	C
	10	100
	8	110
	11	101
	10	101
	11	100
	12	100

Question 1 (4 points) :

Donner, sous forme de tableaux, le schéma et la valeur de la relation construite par chacune des requêtes ci-dessous appliquées aux deux relations R et S (il peut s'agir d'un message d'erreur Oracle).

1. `select C, B
from S where B <> 11;`
2. `select R.A, R.B, S.B, S.C
from R join S on (R.B < S.B);`
3. `select A, count(distinct B) as nbB,
count(C) as nbC
from R natural join S group by A;`
4. `select A, B, count(C) as nbC
from R natural join S group by A;`

Question 2 (2 points) :

Reformuler en SQL chacune des requêtes listées ci-dessous. Suivre très rigoureusement les instructions fournies.

1. `select A, B from R where B in
(select B from S where C > 100);`
2. `select B from S group by B having
count(distinct C) = 1;`

Instruction : la solution ne doit pas utiliser l'opérateur `in` ni sa négation `not in`.

Instruction : la solution ne doit pas utiliser l'opérateur `group by`.

2 A propos de théâtre

On reprend le sujet Théâtre vu pendant le TP2, puis on l'étend afin de prendre en compte les extensions liées aux configurations, aux réductions et à l'utilisation d'une autre salle.

2.1 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).

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). Chaque jour, zéro ou une représentation est donnée.

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 billet 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 billets.

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

2.2 Extensions

Une salle supplémentaire. Une salle supplémentaire a été acquise. Toutes les représentations d'un même spectacle se déroulent dans la même salle. Chaque jour, dans chaque salle il y a zéro ou une représentation.

Les configurations de la salle. Selon les spectacles, toutes les places de certaines zones de la salle où se déroulent ses représentations sont retirées de la vente : en effet, dans certaines mises en scène, une partie des tribunes est occupée par le spectacle. La configuration de la salle, c'est-à-dire l'ensemble des zones dont les places sont mises à la vente, est la même pour toutes les représentations d'un même spectacle.

Les réductions. Pour certaines représentations, le prix des places fait l'objet d'une promotion. Ainsi, à chaque représentation est associé un taux de promotion.

Les personnes qui achètent des places peuvent être des spectateurs ordinaires ou des adhérents et bénéficier d'un tarif réduit. Certaines personnes (étudiants, scolaires, militaires et seniors) bénéficient aussi d'une réduction. Le justificatif de la réduction obtenue est demandé à la personne détenant le billet, à l'entrée du spectacle. Ainsi le prix d'une place fait éventuellement l'objet d'une réduction qui dépend de son type (adhérent, senior, étudiant, sans réduction, etc.). A chacun de ces types est associé un taux de réduction. Le cas échéant, la réduction vient se soustraire au montant remis de la représentation.

2.3 Schéma de relations

LesCatégories (nomC, prix)

/ <c, p> ∈ LesCatégories ⇔ p est le prix des places de catégorie c. */*

LesZones (noZone, noSalle, nomC)

/ <z, s, c> ∈ LesZones ⇔ dans la salle s, les places de la zone z sont dans la catégorie c. */*

LesPlaces (noPlace, noRang, noSalle, noZone)

/ <p, r, s, z> ∈ LesPlaces ⇔ la place de numéro p dans le rang r de la salle s est dans la zone z de la salle s. */*

LesSpectacles (noSpec, nomS, noSalle)

/ <n, t, s> ∈ LesSpectacles ⇔ le spectacle de numéro n a pour titre t. Toutes ses représentations ont lieu dans la salle s. */*

LesConfigurations (noSalle, noZone, noSpec)

/ <s, z, n> ∈ LesConfigurations ⇔ dans la salle s, les places de la zone z peuvent être vendues pour les représentations du spectacle n. */*

LesReprésentations (dateRep, noSpec, reduc)

©M.-C. Fauvet - 2014/15

/ <d, n, r> ∈ LesReprésentations ⇔ une représentation pour le spectacle n est prévue à la date d. Pour cette représentation tous les billets sont vendus avec une réduction de r%. */*
 LesBillets (noSerie, noSpec, dateRep, noPlace, noRang, dateEmission, noDossier, reduc)
/ <t, s, d, p, r, e, n, m> ∈ LesBillet ⇔ le billet 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. Le prix de ce billet bénéficie d'une réduction m%. */*
 LesDossiers (noDossier, montant)
/ <d, m> ∈ LesDossiers ⇔ m est le montant total des places associées au dossier d. LesDossiers est une vue. */*
 domaine (dateRep) = date(heure) */* par ex. 24/11/2007 20H */*
 domaine (dateEmission) = date (seconde) */* par ex. 24/11/2007 9H30MN14S */*
 domaine (nomC) = {'orchestre', 'balcon', 'poulailler'}
 domaine (nomS) = chaînes de caractères
 domaine (noZone) = domaine (noPlace) = domaine (noSalle) = domaine (noSalle)
 = domaine (noRang) = domaine (noSpec) = domaine (noDossier) = entier > 0
 domaine (prix) = domaine (montant) = réels ≥ 0
 domaine (reduc) = réels sur [0, 100[

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).
- Le nombre de places liées à une configuration est ≥ 70
- Chaque jour, dans chaque salle il y a 0 ou 1 représentation
- Dans la relation LesBillets : dateEmission < dateRep
- LesZones[nomC] = LesCatégories[nomC]
- LesPlaces[noZone, noSalle] = LesZones[noZone, noSalle]
- LesSpectacles[noSalle] = LesZones[noSalle]
- LesConfigurations[noSpec] = LesSpectacles[noSpec]
- LesConfigurations[noSalle, noZone] ⊆ LesZones[noSalle, noZone]
- LesReprésentations[noSpec] = LesSpectacles[noSpec]
- LesBillets[noSpec, dateRep, noPlace, noRang] ⊆
 (LesReprésentations * LesConfigurations * LesPlaces)[noSpec, dateRep, noPlace, noRang]
 (* dénote le produit naturel.)
- LesBillets[noDossier] ⊆ LesDossiers[noDossier]

2.4 Expression de requêtes

Indications : la fonction sysdate retourne un instant (à la granularité de la seconde), dont la valeur est observée au moment de son exécution.

to_char(sysdate, 'Month YYYY') = 'Mai 2015'.

to_char(to_date('23/05/2014', 'DD/MM/YY'), 'DD-MM-YY HH:MI:SS') = '23-05-15 12:00:00'

Question 3 (8 points) :

Exprimer en SQL les requêtes ci-dessous.

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 imbriquée sera soigneusement spécifiée.

1. Donner l'expression SQL associée à la vue LesDossiers.
2. Quelles sont les places (numéro du rang et numéro de la place dans le rang) qui ne peuvent pas être vendues (ou réservées) pour le spectacle numéro 10 ?
3. Pour chaque représentation, combien y a-t-il de places libres (c'est-à-dire qui n'ont pas été réservées, et qu'il est encore possible de réserver/d'acheter) ?

Indication : bien qu'intuitive, l'expression SQL select count(A, B)... est incorrecte.

4. Chaque mois, quelle est la représentation (date et numéro du spectacle) pour laquelle le plus grand nombre de billets est ou a été vendu ? (un mois est donné par le nom du mois et le numéro de l'année : par exemple Janvier 2015)

2.5 Implantation de l'application

Nous considérons la base de données telle que spécifiée Section 2.1. Le code SQL permettant de définir les relations sauf `LesSpectacles`, `LesZones`, `LesPlaces` et `LesCategories` est donné en annexe (voir page 4) :

Question 4 (2 points) :

Fournir le code SQL Oracle qui permet de définir les relations `LesReprésentations`, `LesConfigurations` et `LesBillets`.

Question 5 (2 points) :

Parmi toutes les contraintes d'intégrité spécifiées dans l'énoncé (voir section 2.3), lesquelles sont à la charge des applications ?

Question 6 (3 points) :

Le programme `nouveau-spectacle-action.php` écrit en PHP et dont le squelette est donné page 5 en annexe permet d'enregistrer un nouveau spectacle et de construire un message destiné à l'utilisateur qui l'informe de l'issue de cette création. Compléter le programme `nouveau-spectacle-action.php` (utiliser la page fournie avec le sujet).

Indication :

On suppose l'existence de la fonction dont la signature est `getConstraintName: string → string`. Par exemple, soit `e` une erreur SQL retournée par le système, `getConstraintName(e)` est le nom de la contrainte dont la tentative de validation a échoué (par exemple `cat_c3`).

A Code SQL de création de certaines relations de la BD Théâtre

```
create table LesCategories (nomC varchar(20), prix number (6,2) not null,
    constraint cat_c1 primary key (nomC),
    constraint cat_c2 check (prix > 0),
    constraint cat_c3 check (nomC in ('orchestre', 'poulailler', 'fauteuil')) );

create table LesZones (noZone number (4), noSalle number (4), nomC varchar(20) not null,
    constraint zones_c1 primary key (noZone, noSalle),
    constraint zones_c2 foreign key (nomC) references LesCategories (nomC),
    constraint zones_c3 check (noZone > 0), constraint zones_c4 check (noSalle > 0) );

create table LesPlaces (noPlace number (4), noRang number (4), noSalle number (4) not null,
    noZone number (4) not null,
    constraint places_c1 primary key (noPlace, noRang, noSalle),
    constraint places_c2 foreign key (noZone, noSalle) references LesZones (noZone, noSalle),
    constraint places_c3 check (noPlace > 0), constraint places_c4 check (noRang > 0) );

create table LesSpectacles (noSpec number (4), nomS varchar(40) not null, noSalle number (4),
    constraint spec_c1 primary key (noSpec),
    constraint spec_c2 foreign key (noSalle) references LesZones(noSalle),
    constraint spec_c3 check (noSpec > 0) );
```

Merci d'utiliser cette page pour répondre à la question 5.

Numéro d'anonymat :

```

/* Les valeurs des variables noSpec, dateRep, reduc et noSalle ont été saisies par le biais
 * d'un formulaire. */
<?php
include('entete.php');
$noSalle = $_POST['noSalle']; $nomS = $_POST['nomS']; $noSpec = $_POST['noSpec'];
$dateRep = $_POST['dateRep']; $reduc = $_POST['reduc']; $noSalle = $_POST['noSalle'];
// Hypothèse : la relation dummy existe et contient un nuplet
$requete = (" select greatest (sysdate, to_date(:d, 'DD-MM-YYYY')) - sysdate from dummy ");
$curseur = oci_parse ($lien, $requete) ;
oci_bind_by_name ($curseur, ':d', $dateRep);
$ok = oci_execute ($curseur, OCI_NO_AUTO_COMMIT) ;
if (!$ok) {
    echo "Problème d'accès à la base de données. Spectacle non ajouté.";
    oci_rollback($lien);
} else {
    $res = oci_fetch ($curseur); $diff = (double) oci_result ($curseur, 1);
    if ($diff<=0) {
        echo ("Il est impossible de créer une représentation dans le passé. Spectacle non ajouté.") ;
        oci_rollback($lien);
    } else {
        $reqInsert = (" insert into LesSpectacles values (:n, :s, :a) ");
        $reqInsertRep(" insert into LesRepresentations values (to_date(:d, 'DD-MM-YYYY'), :n, :r) ");
        $curInsert = oci_parse($lien, $reqInsert); $curInsertRep = oci_parse($lien, $reqInsertRep);
        .....
        .....
        .....
        .....
        .....
        $ok = oci_execute($curInsert,OCI_NO_AUTO_COMMIT);
        if (!$ok) {
            $error_code = oci_error($curInsert);
            $constraint = getConstraintName($error_code);
            switch ($constraint) {
                case "spec_c1": $message = "Le numéro du spectacle est déjà affecté." ; break;
                default:        $message = "Problème avec la base de données"; break;
            }
            echo $message ; echo "Spectacle non ajouté.";
            .....
        } else {
            $ok = oci_execute ($curInsertRep, OCI_NO_AUTO_COMMIT) ;
            if (!$ok) {
                $error_message = oci_error($curInsert);
                $constraint = getConstraintName($error_code);
                switch ($constraint) {
                    case ".....": $message = "....."; break;
                    case ".....": .....
                    .....
                    .....
                    default: $message = "Problème avec la base de données"; break;
                }
                echo $message; echo "Spectacle non ajouté";
                .....
            } else {
                echo "Spectacle ajouté." ;
                .....
            }
        }
    }
}
}
}

```