

Blocs Anonymes, procédures, fonctions, curseurs implicites et explicites

CUSTOMER_CST => Données client

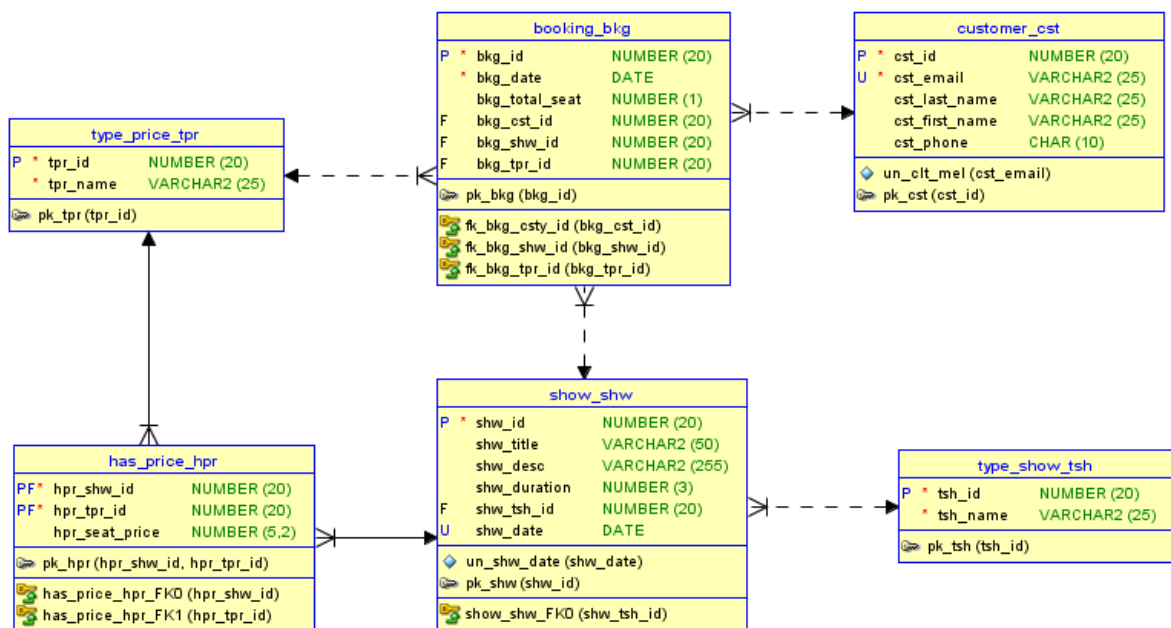
HAS_PRICE_HPR => Le prix d'une place de spectacle dépend du spectacle et du type de tarif

SHOW_SHW => Données spectacle

TYPE_PRICE_TPR => Type de tarif

TYPE_SHOW_TSH => Type de spectacle

BOOKING_BKG => Données de réservation. Bkg_total_seat est le nombre de places réservées pour un client et un spectacle, avec un type de tarif



1. Bloc Anonyme – JOIN

Créer un bloc anonyme qui affiche le nom du client ayant réalisé une réservation particulière. Testez votre bloc en saisissant au clavier le numéro de réservation.

2. Procédure – INSERT

Version 1

Créer une procédure permettant d'insérer une nouvelle réservation pour un spectacle donné. Les paramètres sont : numéro de spectacle, nombre de places, type de tarif) Utiliser « current_date » obtenir une valeur de date. Le nombre de places vaut 1, de même que le type de tarif. Le numéro de client sera fixé à une valeur de votre choix. Le numéro de spectacle pourra être saisi au clavier.

On suppose que la procédure n'a pas à gérer le cas où le numéro de spectacle (ou le type de tarif...) n'est pas correct.

Avant l'insertion de la nouvelle ligne, vous afficherez le nombre de réservations déjà réalisées pour ce spectacle ainsi que le nombre de places réservées.

Version 2

Changez votre bloc pour que la nouvelle valeur de réservation soit générée par une séquence.

Créez votre séquence avec l'instruction :

```
CREATE SEQUENCE bkg_seq  
START WITH xxxxx  
INCREMENT BY 1  
NOCACHE ;
```

Remplacez xxxxx par la valeur qui devra être générée en premier par la séquence. Cette instruction est une instruction de « DDL ».

La valeur suivante dans une séquence est produite par `nom_de_sequence.NEXTVAL`

Quelle version vous semble la meilleure ? Pourquoi ?

3. Procédure – DELETE

Ecrire une procédure permettant de supprimer une réservation. Votre bloc permettra d'afficher un message approprié selon le succès ou l'échec de la suppression.

4. Procédure – UPDATE

Ecrire une procédure permettant d'augmenter les prix des spectacles d'un certain coefficient. Seuls les spectacles pour lesquels il n'y a aucune réservation seront mis à jour. Votre bloc affichera le nombre de lignes mises à jour.

5. Bloc anonyme - Curseur explicite

Créer un bloc anonyme qui affiche les détails des réservations, triés par nom de client. Les détails incluent : booking id, date et nombre total de places, prénom et nom des clients, titre du spectacle.

6. Curseur explicite

Créer une procédure qui réduit les prix des places (`hpr_tpr_id`). Le coefficient de réduction (supposé positif) est passé en paramètre.

La procédure ne doit pas remettre en cause la hiérarchie des types de prix : les prix au tarif 1 doivent rester supérieurs à ceux du tarif 2, et ces derniers à ceux du tarif 3. Les prix de tarif 3 ne peuvent pas être modifiés.

Le nombre des lignes mises à jour sera affiché.

Vous devez vous assurer qu'aucune autre transaction ne modifie les données avant que votre traitement ne soit terminé. Votre traitement ne devra pas être mis en attente.

7. Curseurs multiples avec paramètres – GROUP BY

Créer une procédure qui calcule la moyenne des durées de spectacle par type de spectacle, et affiche, pour chaque type de spectacle, la liste de ceux qui ont une durée supérieure à la moyenne

La résultat devra être semblable à :

THÉÂTRE 85	=> <i>type of show – average duration</i>
Show title : CALACAS - ZINGARO 90	=> <i>show title of type Theatre – duration > computed avg</i>
Show title : LES AVEUGLES 90	=> <i>show title of type Theatre – duration > computed avg</i>

Et ainsi de suite.