

R2.06-Exploitation BD (Séance n° 1)

Opérateurs ensemblistes, Manipulation de la structure de la base, Mises à jour et Transactions

Objectifs du TP :

L'objet de cette première séance de TP est de rappeler les jointures, de prendre en compte différentes évolutions de l'univers réel modélisé par la base de données exemple en effectuant diverses manipulations sur la structure de cette base (*intension de la base*). Il s'agit, également, d'utiliser les opérations ensemblistes et les opérateurs de mises à jour des données (*extension de la base*) et illustré le mécanisme de transaction.

Rappel du schéma relationnel de la base Voyages :

VOYAGE	(<u>IDV</u> , VILLEARR, PAYSARR, VILLEDEP, HOTEL, NBETOILES, DUREE)
PLANNING	(<u>IDV</u> , DATEDEP, TARIF)
CLIENT	(<u>NUMCL</u> , NOM, PRENOM, ADRESSE, CP, VILLE, CATEGORIE)
RESERVATION	(<u>NUMCL</u> , (<u>IDV</u> , <u>DATEDEP</u>), NBPERS, DATERES)
OPTIONV	(<u>CODE</u> , LIBELLE)
CARAC	(<u>IDV</u> , <u>CODE</u> , PRIX)

Première étape : Expression des jointures - Rappel

Formulez les requêtes suivantes en utilisant, si possible, les **trois** formes de jointure.

- Q1 : Donnez, les voyages (date de départ, ville d'arrivée et tarif) à destination du Maroc.
- Q2 : Donnez la liste des réservations (date de départ, ville d'arrivée et pays arrivée du voyage) des clients n'habitant ni Paris ni Marseille.
- Q3 : Quelles sont les options (leur libellé) gratuites proposées pour les voyages réservés par le client Nicolas Barbier ?
- Q4 : Quels sont les clients (nom, prénom, ville) ayant réservé un voyage à destination d'Istanbul et partant de leur ville de résidence ?

Deuxième étape : Utilisation des Opérateurs ensembliste et équivalence

Formulez les requêtes suivantes *en faisant appel qu'aux opérateurs ensemblistes*.

- Q5 : Quelles sont les villes de départ d'un voyage dans lesquelles résident des clients ?
- Q6 : Quel est le libellé des options communes aux voyages 354 et 952 ?
- Q7 : Existe-t-il des voyages, si oui, donnez-leur identifiant et destination, pour lesquels il n'y a aucune réservation ?
- Q8 : Quelles sont les options (libellé) gratuites pour le voyage 354 et celles qui sont payantes pour le voyage 952 ?

Formulez les requêtes suivantes *en ne faisant pas appel aux opérateurs ensemblistes*.

- Q9 : Quels sont les voyages (identifiant et destination) offrant à la fois les options 'Visite guidée' et 'Piscine'.
- Q10: Existe-t-il des clients, si oui, donnez leur nom et prénom, qui n'ont aucune réservation ?

Troisième étape : Modifications de la structure de la base

- Q11: On souhaite intégrer dans la relation `PLANNING` un deuxième tarif de voyage pour les enfants. Effectuer l'ajout de cet attribut `TARIFENF` dans la relation en utilisant le même type de données que pour `TARIF`. De manière similaire, le nombre d'enfants doit être ajouté dans la relation `RESERVATION`. Donnez à cet attribut `NBENF` le même type que celui de `NBPERS`.
- Q12: On souhaite étendre la définition du type de l'attribut `LIBELLE`, en augmentant la taille des valeurs admissibles de 10 caractères. Consultez la définition de cet attribut puis effectuez sur la base la modification correspondante et vérifiez-la.
- Q13 : On souhaite intégrer dans la base une contrainte de domaine pour l'attribut `CATEGORIE` qui prendra ses valeurs dans l'ensemble suivant : {PRIVILEGIE, BON, MAUVAIS}. Effectuez cet ajout de contrainte.
- Q14: Spécifiez une contrainte de domaine pour l'attribut `NBETOILES` dans `VOYAGE` afin que ses valeurs appartiennent à l'intervalle [1 .. 5].
- Q15: On souhaite intégrer dans la base une nouvelle relation permettant de connaître par hôtel et type de chambre, la capacité offerte (nombre de chambres). Le schéma de la relation est le suivant :

`CAPACITE (HOTEL, TYPEC, NBCH)`

Créez cette nouvelle relation sans oublier que `HOTEL` doit avoir le même type que dans `VOYAGE`. Spécifiez la contrainte de clef primaire ainsi que la contrainte de domaine suivante : le type des chambres doit appartenir à l'ensemble {Single, Double, Double luxe,

Suite, Suite junior, Suite prestige}.

Quatrième étape : Mises à jour des données

Sur la base modifiée, on vous demande de réaliser les opérations de mise à jour des données suivantes :

Q16 : Mettez à jour l'attribut `NBENF` en lui affectant la valeur 2 pour toutes les réservations du client 2103. Modifiez cet attribut en lui donnant la valeur 1 pour les réservations du client Thomas Jarolim.

Q17: Pour tous les tuples de la relation `PLANNING`, effectuez la modification de `TARIFENF` en lui affectant la valeur de `TARIF` réduite de moitié.

Q18: Insérez deux nouveaux tuples dans la relation `CLIENT` en vérifiant la mise en œuvre de la contrainte de domaine sur `CATEGORIE`. Créez une réservation pour un de ces clients et un voyage existant.

Q19: Mettez à jour la relation `VOYAGE` en tentant de donner une valeur invalide à l'attribut `NBETOILES`.

Q20: Créez les tuples suivants dans la relation `CAPACITE`.

HOTEL	TYPEC	NBCH
ANTIQUE	SINGLE	10
ANTIQUE	DOUBLE	75
ANTIQUE	DOUBLE LUXE	12
ANTIQUE	SUITE	5
ELIAS BEACH	DOUBLE	83
ELIAS BEACH	SUITE	27
OLD BRIDGE	SINGLE	25
OLD BRIDGE	DOUBLE	75
SAFARI JAMBO	SINGLE	32
SAFARI JAMBO	DOUBLE	100
TRANSATLANTIQUE	DOUBLE	200
BAMBURI	DOUBLE	150

Cinquième étape : Archivage d'information et gestion de transactions

Le mécanisme des transactions est ici illustré à travers un exemple d'archivage d'information.

Dans un souci d'archivage des données, on désire régulièrement “ vider ” la relation `RESERVATION`, sans pour autant perdre les informations existantes.

Q21: Créez la relation une nouvelle relation appelée `ARCH_RESERV`, dans laquelle seront “ rapatriés ” tous les tuples décrivant les réservations pour des voyages programmés passés. La relation `ARCH_RESERV` doit comporter les mêmes attributs que `RESERVATION` mais au lieu de conserver la date de réservation, on gardera la date d'archivage (nouvel

attribut DATEARCH). Spécifiez seulement la contrainte de clef primaire.

Q22: Il s'agit, à présent de définir l'ensemble des requêtes permettant d'effectuer l'archivage des réservations dans ARCH_RESERV.

Pour réaliser les différentes opérations nécessaires (décrites plus loin), vous procéderez de la manière suivante :

- Créez la table RESERV_SAUV à partir de la table réservation.
- Créez un fichier archive.sql, les différentes requêtes seront dans ce fichier séparées par une ligne réduite au caractère /. Ceci permet à Oracle, lors du chargement du fichier sous sqlplus, d'enchaîner automatiquement l'exécution des différentes requêtes.
- Procédez pas à pas, i.e. en rajoutant au fichier archive.sql une nouvelle requête seulement après avoir testé l'exécution des précédentes, en consultant l'extension de la relation ARCH_RESERV (et celle de RESERV_SAUV pour la dernière suppression). Évidemment, après cette vérification, il vous faudra, pour pouvoir relancer le fichier sans problème, restaurer la base dans l'état précédant toutes les mises à jour de données (i.e. dans son état au dernier **COMMIT**), en utilisant l'ordre **ROLLBACK**.

Les différentes opérations à réaliser sont les suivantes :

- Insertion automatique, dans ARCH_RESERV, de toutes les réservations dont la date de départ date de plus de trois mois (ou plus de 90 jours entre cette date et la date du jour) et affectation automatique de la date du jour pour l'attribut DATEARCH.
- Suppression de tous les tuples de RESERV_SAUV qui **viennent d'être insérés** dans ARCH_RESERV.

Lorsque toutes les manipulations sont au point, faites en une transaction.