**Examen SGBD**

**Pour assurer la cohérence d'une base de données, on évite la redondance d'information : chaque donnée est stocké une seule fois dans la base.**

**Il arrive cependant que l'on accepte certaines redondances.**

1. **Expliquez dans quel but**
2. **Que fait-on pour éviter que ces redondances provoquent des incohérences dans la base ?**
3. **Comment qualifie-t-on cette redondance ?**
4. **Donnez un exemple (pas besoin de code expliquez le principe)**
5. Dans le but de satisfaire à des besoins de vitesse de traitements, et de ce fait gagner du temps de réponse. *Améliorer vitesse de traitement et temps de réponse*

B. On utilise des Triggers où déclencheurs pour éviter les incohérences

C. C'est la redondance contrôlée

D. CREATE TRIGGER maj\_nb\_emp

BEFORE INSERT num\_service

ON personnel

FOR EACH ROW

BEGIN

**IF** inserting **THEN**

**UPDATE** service **SET** nombre\_emp = nombre\_emp + 1

Where num\_service := NEW.num\_service;

**END IF;**

**END;**

Une fois qu'une commande du LDD s'exécutent, il est possible de définir des déclencheurs qui réagissent à ces commandes

Par exemple générer automatiquement une valeur de clé primaire, Gérer automatiquement la redondance

**Dans les bases de données relationnelle, on parle de "catalogue"**

1. **Donnez 2 synonymes**
2. **Par qui est-il mis à jour ?**
3. **Où est-il stocké ?**
4. **Que contient-il ?**
5. Dictionnaire, méta-base
6. Par le SGBD, mis automatiquement à jour par le SGBD.
7. Stocké dans la base de données BD elle-même.
8. Il comporte un ensemble de tables, vues et contraintes. **Contient** la description de tous les objets présents dans la base : tables, domaines, contraintes, privilèges, fichiers, index, procédures stockés, déclencheurs, vues, objets. **Constitue** une mini base de données

**Les différents niveaux d'une base de données :**

1. **Citez les différents niveaux dites ce qu'ils contiennent**
2. **Citez les différents niveaux d'indépendances expliquez à quoi ils correspondent**
3. - **Niveau Externe** : Vues, spécifications de procédures.

- **Niveau Logique** : Tables, implémentations de procédure, Déclencheurs.

- **Niveau Physique** : Contient les index, Clusters, Fichiers.

1. **- Indépendance Niveaux LOGIQUE** : on peut changer la structure logique globale sans devoir changer les programmes d'applications.

**- Indépendance Niveaux PHYSIQUE :** signifie que la couche physique et l'organisation des données peuvent changer sans devoir changer la structure logique globale ou les programmes d'applications. *En clair on peut modifier la disposition physique (fichiers) sans modifier le niveau logique et donc le niveau externe*

|  |  |  |
| --- | --- | --- |
| **Question** | V | F |
| Une exception définie "non redirigée" doit être déclarée par le programmeur  Dans la section des déclarations d'un bloc PL/SQL. | **V** |  |
| Une exception définie "redirigé" doit faire l'objet d'une instruction "PRAGMA EXCEPTION\_INIT" par le programmeur dans la section des déclarations d'un bloc PL/SQL. | **V** |  |
| En PL/SQL, dans une clause WHEN de la section EXCEPTION, il est possible de "relancer" une nouvelle exception. Je sais pas pour WHEN OTHERS oui intercepter n'importe quel type d'exceptions | X | X |
| Une exception prédéfini doit être lancée explicitement |  | **F** |
| Une exception défini "non redirigée" n'est pas déclenchée automatiquement par Oracle | **V** |  |
| Toutes les erreurs détectées par Oracle sont des exceptions prédéfinies | **V** |  |
| En PL/SQL, il est possible d'écrire des procédures polymorphes | **V** |  |
| En PL/SQL, par défaut, le passage de paramètre s'effectue par adresse \* |  | **F** |
| Une collection est un ensemble , éventuellement ordonné, d'éléments de types différents |  | **F** |
| Une collection de type "Tableau associatif " peut posséder des indices non-consécutifs à vérifier mais presque sur | **V** |  |
|  |  |  |

\*Par défaut les paramètres OUT et IN OUT sont passés **par valeur.** Les valeurs sont donc copiées avant l'exécution du sous-programme

**Peut-on dire que le PL/SQL est un langage orienté objet ? oui, non, pourquoi ?**

PL/SQL est un langage procédural. Cependant des fonctionnalités OO existent. PL/SQL à une forte coloration OO. Pourquoi du fait de l'utilisation d'objet.

De plus l'intégration de déclencheurs et de procédures stockées favorise l'usage du PL/SQL dans un paradigme de programmation orienté objet.

On parle de polymorphisme parce que l'on peut créer des fonctions semblables de nom mais avec des paramètres différents.

Les objets en PL/SQL table et vue sont considéré comme des objets

**Question**

**Quel est le résultat d'exécution (les affichages réalisés) par le bloc PL/SQL suivant ?**

****

L'affichage donnera

BLOC1

BLOC2ERR

BLOC1ERR

**Quelle est la clause qui permet de modifier un tuple tout en retournant des données de ce tuple ?**

**Returning into**

**Quelle est la directive de compilation qui permet de passer des paramètres en références en PL/SQL ?**

La directive **NOCOPY,** [permet de passer les paramètres OUT et IN OUT par reference.

**Ecrivez la commande permettant de déclarer l'association du code d'erreur -2291 à l'exception ExcIntRef ?**

****

**Réécrivez le code suivant en donnant des valeurs par défaut à chacun des paramètres ('Dupont' pour le nom, 4000 pour le code postal et 100 pour le crédit)**

****

**Quel est le résultat d'exécution (les affichages réalisés) par le bloc PL/SQL suivant ?**

****

L'affichage donnera lieu à : Réponse 3

3) Nom1 <> Nom2

Nom2 = Nom3

**Soient les tables suivantes**

****

Le trigger DepartementsUpd :

1. S'éxécutera lors de la mise à jour du Deptno dans la table employes
2. Est invalide, il est impossible de modifier une clé étrangère dans un déclencheur
3. Est invalide car il n'y a pas de COMMIT, après la commande UPDATE
4. Est invalide car la variable :OLD ne peut être référencée dans un UPDATE

Réponse 7 Aucune réponse proposée n'est correcte

**Soit le déclencheur suivant défini sur la table Clients**

****

**Réponse 3**. BEFORE BEFORE BEFORE

**Soit le déclencheur suivant défini sur la table clients**

****

Réponse : je ne sais pas !

**Pour traiter une erreur définie "redirigée", le programmeur doit**

1. **La déclarer dans la section des déclarations**
2. **Utiliser le pragma exception\_init pour associer l'exception à un code d'erreur oracle**
3. **La définir dans le gestionnaire d'exception du bloc**

Réponse : [ 2 ]

**Quelle est l'exception prédéfinie qui est déclenchée lorsqu'une instruction SELECT INTO retourne plus d'une ligne ?**

1. **TOO\_MANY\_ROWS**
2. **NO\_DATA\_FOUND**
3. **DUP\_VAL\_ON\_INDEX**
4. **SQL%NOTFOUND**

Réponse : [ 1 ] TOO\_MANY\_ROWS

**Quelle clause spécifiée dans un gestionnaire d'exceptions permet d'intercepter n'importe quelle exception ?**

1. **WHEN OTHERS**
2. **WHEN NO\_SPECIFIED**
3. **WHEN NULL**
4. **LA DERNIERE CLAUSE SPECIFIEE DANS LE BLOC D'EXCEPTIONS**

Réponse : [ 1 ]

**Quelle est l'exception prédéfinie qui est déclenchée lorsqu'une instruction**

**SELECT … BULK COLLECT … INTO … ne retourne pas de ligne**

1. **TOO\_MANY\_ROWS**
2. **NO\_DATA\_FOUND**
3. **DUP\_VAL\_ON\_INDEX**
4. **SQL%NOTFOUND**

Réponse : Je dis que c'est la 2 mais la prof avait barré la réponse du gars mais j'ai essayé la requête

**Quelle est l'exception prédéfinie qui est déclenchée lorsqu'un curseur ne retourne pas de ligne ?**

1. **TOO\_MANY\_ROWS**
2. **NO\_DATA\_FOUND**
3. **DUP\_VAL\_ON\_INDEX**
4. **SQL%NOTFOUND**

Réponse : [ 4 ], car %NOTFOUND vaut true si un fetch n'a pas pu extraire de tuple

**Soit le code suivant, quel sera le résultat de son exécution ?**

****

1. **Hurray sera affiché**
2. **Rien ne sera affiché, le contrôle sera simplement rendu à l'environnement d'appel**
3. **Une exception NO\_DATA\_FOUND sera lancée**
4. **Une exception VALUE\_ERROR sera lancé**

Réponse : [ 3 ], j'ai testé

**Analysez le code du déclencheur ci-dessous. Expliquez son rôle. Est-il possible de le rendre plus efficace en améliorant le code ? Si oui, faites les modifications nécessaires.**

****

C'est un trigger où déclencheurs qui à chaque update du salaire des membres du personnel va comparer l'ancien salaire avec le nouveau salaire si le nouveau est plus petit que l'ancien salaire une exception est lancé. L'exception salaire\_diminue qui affichera le message "Le salaire ne peut diminuer".

Oui nous pouvons rendre le code plus efficace.

****

---------------------------------------------------------------------------------------------------------------------------

**Analysez le code ci-dessous. Expliquez ce qu'il réalise.**

**Modifier le code de manière à obtenir le même résultat sans utiliser le BULK COLLECT INTO …**

**Est-il possible de le rendre plus efficace en améliorant le code**

On utilise un curseur qui permet de traiter les requêtes de recherche dont le résultat contient un nombre quelconque de tuple. Ici le curseur est utilisé pour parcourir l'ensemble des tuples de la table Employes avec comme paramètre de recherche where job like 'A%'. Cette expression de sélection est spécifiée dans sa définition. Une boucle for est utilisé pour affiché les employées répondant aux critères de recherche. On remarque bien les 3 étapes nécessaires aux curseur. 1)La déclaration 2)L'ouverture OPEN 3) La fermeture CLOSE

****

Peut-on améliorer, ça dépend par exemple à la place d'utiliser le FOR on peut utiliser l'instruction while

1. **Analysez le code ci-dessous. Expliquez ce qu'il réalise (soyez précis)**
2. **Donnez la portion de MRD correspondant à ce code**
3. **Quel est le type de contrainte utilisé à la ligne 6; à la ligne 17 ? Dans quel cas est-il nécessaire d'utiliser le type de contrainte de la ligne 17 ?**
4. **Aurait-il été possible de donner l'ensemble de ces instructions dans un ordre différent ? Expliquez**
5. **Quel est l'intérêt de donner un nom à chacune des contraintes ? Expliquez**

****

1. Ligne 1 & 2 si les tables Coureur et Club existe elles seront supprimées. Si elle n'existe pas une erreur sera affiché table or view does not exist. En utilisant la clause **CASCADE,** l'effacement de la table entraîne automatiquement la suppression de tous les objets qui lui faisaient référence (index, vue, contrainte, déclencheur)

L3 création de la table **Coureur.**

L 4,5 & 6 création d'un attribut IdCoureur qui sera clé primaire cette contrainte porte un nom

Création d'un attribut Nom de type varchar2 ce nom ne peut pas être null, et il doit être unique

Création d'un attribut date de Naissance ne peut pas être null

Création attribut Sexe ne peut être null et la contrainte check est là pour obliger à vérifier si le Sexe est bien soit M ou soit F

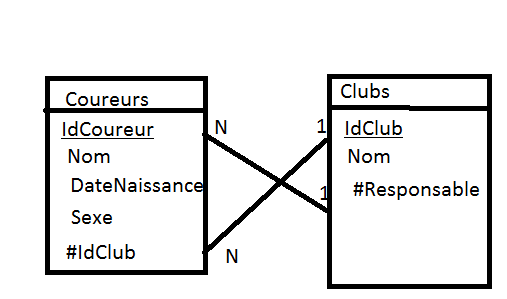
15 & 16 Ajout d'un attribut idClub contrainte non null

L21 22 23 : creation d'un table Club avec un attribut idClub + contrainte clé primaire

24 25 ajout d'un attribut nom ave contrainte non null

26 27 28 Ajout d'un attribut responsable avec comme contrainte une référence vers un coureur (donc référence à la table coureur).

L30 31 32 on modifie la table coureur, on va mettre une contrainte sur l'attribut IdClub. C'est une contrainte de clé étrangère on fait une référence vers un club (table club). Double dépendance entre la table club et la table coureur.



1. Ligne 6 : PRIMARY\_KEY Spécifie que la colonne est définie comme une clé primaire. Même si l'effet est le même que pour UNIQUE, aucune valeur null n'est accepté !!!

Ligne 17 C'est une contrainte de type check oblige à vérifier la condition associée pour chaque valeur rentrée

1. Non parce que par exemple la table club à besoin de l'identifiant IdCoureur de la table Coureur donc on ne peut pas changer l'ordre même pour le alter table on ne pourra pas changer l'ordre si on change l'ordre nous aurons des erreurs
2. Bonne pratique de programmation car cela permet d'avoir une meilleure lisibilité. Lorsque l'on viole une contrainte, oracle va nous prévenir avec une note d'erreur si les contraintes ont été nommé. Sinon oracle va sortir la contrainte violée avec un numéro ce qui n'est pas très pratique !

**Comment le SGBD exécute-t-il**

1. **Des requêtes (simplement) imbriquées**
2. **Des requêtes corrélées ?**

**Vous pouvez éventuellement utiliser un exemple pour étayer votre explication !**

Pour les requêtes imbriquées le système commence par évaluer le bloc interne.

Les requêtes imbriquées, fait en sorte que la requête principale donne à la sous-requête de nouvelles conditions d'exécution à chaque fois que la requête principale trouve une ligne correspondant à ses propres filtres.

**Pour les requêtes corrélées**. La sous question référence le niveau externe. Dans ce type de requêtes, il est impossible d'obtenir le résultat de la sous-question sans faire référence au niveau externe

****

*Il existe des requêtes pour lesquelles la sous question référence le niveau externe. Dans ce type de requêtes, il est impossible d'obtenir le résultat de la sous-question sans faire référence au niveau externe. On parle de sous questions corrélatives.*

*Une requête corrélée est une sous-requête qui s'exécute pour chaque ligne de la requête principale et non une fois pour toute. C'est comme un boucle imbriquées.*

|  |  |  |
| --- | --- | --- |
| **Question** | **V** | **F** |
| En Oracle, une lecture bloque une modification et une modification bloque une lecture |  | **X** |
| En Oracle, une écriture bloque une écriture, il est donc impossible de provoquer une perte de mise à jour \* |  | **X** |
| En oracle le niveau d'isolation par défaut est REPEATABLE READ |  | **X** |
| Le niveau d'isolation REPEATABLE READ peut produire des lectures impropres lors de l'exécution de transaction concurrentes |  | **X** |
| Par défaut, en Oracle, il est possible d'obtenir des lectures non reproductibles et des lectures impropres lors de l'exécution de transactions  concurrentes |  | **X** |

**\*\*Les différentes transactions sont constituées d'actions lire et écrire et, si aucun mécanisme de contrôle de l'exécution n'est présent dans le SGBD, nous pouvons avoir les 3 types d'anomalies :**

* **Pertes de mise à jour**
* **Lecture impropre**
* **Lecture non reproductible**