

SAE MS204 – SEMAINES 20 et 21

Date de début de la SAE : lundi 15 mai 2023 8H15

Date de clôture de la SAE : vendredi 26 mai 23h59

Le département du pas de Calais recense les différents sites touristiques de son département qui proposent un certain nombre d'activités de loisirs. En outre, tout au long de l'année, et plus particulièrement en période estivale, des événements sont organisés sur certains de ces sites auquel tout le monde peut assister dans la limite des places disponibles. Les personnes désirant participer doivent s'inscrire pour réserver le nombre de places désiré.

1. Création de la base TOURISME62

Voici le modèle relationnel obtenu après analyse.

THEME (cdTheme, libThme)

TERRITOIRE (cdTerr, nomTerr)

SITE (cdSite, nomSite, tpSite, adrSite, cpSite, villeSite, emailSite, telSite, siteweb, cdTerr#, cdTheme#)

EVENEMENT (cdSite#, numEv, dateDebEv, dateFinEv, ndPlaces, tarif)

PARTICIPANT (cdPers, nomPers, prenomPers, adrPers, cpPers, villePers, telPers, tpPers)

RESERVATION (cdSite#, numEv#, cdPers#, dateResa, nbPlResa, modeReglt)

Remarques : le mode de règlement est un entier (1 pour Carte Bancaire, 2 pour Chèque et 3 pour Espèces) et le type de participant est donné sur un seul caractère (P,C ou E).

- a) Retrouver le MCD dont est issu ce MLD et le réaliser sur power AMC (créer un fichier **Tourisme62.mcd**).
- b) Définir les types de données : Choisir les types de données les mieux adaptés à chaque colonne (Choisir le type entier pour toutes les clés primaires sauf le champ `cdAct` (CHAR 1) et le champ `dateResa` bien sûr) et définir les contraintes NOT NULL nécessaires. Les numéros de téléphone sont à définir sur 10 caractères.
- c) Ajouter les contraintes suivantes :
 - Le nombre places (`nbPlaces`) pour un événement est toujours supérieur à 20 ;
 - Le mode de règlement est compris entre 1 et 3 ;
 - Le type participant est 'P' (Particulier), 'C' (Collectivité) ou 'E' (Etudiant)
- d) Générer le MPD (pour ORACLE v11) puis le script de création des tables (**crebas_Nom1_Nom2.sql**) et faire en sorte que :
 - Les contraintes soient générées lors de la création des tables (et non après coup dans des ALTER TABLE).
 - Aucun index ne soit généré, ni aucune commande de suppression de clés étrangères.
- e) Dans le script ainsi généré, vérifier la génération des contraintes et appliquer les modifications suivantes :
 - Ajouter la contrainte suivante : La date de fin d'événement peut être nulle ou doit être supérieure ou égale à la date début de l'événement.
 - Supprimer du script généré les contraintes NOT NULL qui vous semblent inutiles.
- f) Tester la création de la base sous ORACLE.

2. Modification du script de création de la base.

On désire stocker également une liste d'activités proposées par les différents sites répertoriés :

ACTIVITE
<u>cdAct</u>
nomAct

PROGRAMME
<u>cdAct</u>
<u>cdSite</u>
tpPublic

La table **PROGRAMME** permet de savoir quelle activité est proposée par quel site en précisant le type de public concerné.

Le type de public peut prendre les valeurs suivantes : **TOUS** pour une activité proposée à tout public, **+18** pour une activité réservée aux plus de 18 ans, **+10** pour une activité pour les plus de 10 ans, **+5** pour une activité uniquement pour les plus de 5 ans.

Compléter le script précédemment généré en ajoutant :

- Les commandes de création et de modification nécessaires pour la création des tables **ACTIVITE** et **PROGRAMME** en appliquant l'intégrité référentielle. Contrairement à la table **PROGRAMME**, créée vide, la table **ACTIVITE** sera créée à partir de la table **TESTSAELD.ACTIVITE** en récupérant les données.
- La directive **ON DELETE CASCADE / ON DELETE SET NULL** dans les **CREATE TABLE** où cela semble judicieux.
- Une colonne **dateNais** dans la table **PARTICIPANT**
- Une colonne virtuelle **dureeEv** (donnant la durée de l'événement en nombre de jours) dans la table **EVENEMENT**.
- Une contrainte de domaine de votre choix (ayant un intérêt)

Colonnes virtuelles

On peut créer dans une table une **colonne virtuelle** dont la particularité est de ne pas être stockée mais évaluée automatiquement à la demande (lors de son utilisation dans une requête, ou une instruction de mise à jour) à partir d'une expression qui permet de déterminer sa valeur. Une colonne virtuelle ne peut pas faire référence à une autre colonne virtuelle.

Exemple :

```
CREATE TABLE PILOTE (  
    numPil CHAR(4) PRIMARY KEY,  
    nomPil VARCHAR2(30) NOT NULL,  
    salaire NUMBER(8,2) NOT NULL,  
    salaireAnnuel GENERATED ALWAYS AS (salaire*12) VIRTUAL);
```

Dans cet exemple, la colonne **salaireAnnuel** est une colonne virtuelle générée à partir du **salaire** et ne sera recalculée que si elle est utilisée.



- Ajouter les index sur
 - Les clés étrangères,
 - Le nom des sites
 - Les noms-prénoms des participants
 - Les noms des activités.

Relancer la création de la base sous ORACLE.

3. Chargement de la base Vidéothèque

a) Tables **THEME** et **TERRITOIRE**

Ecrire les commandes d'insertions (à la suite toujours dans le même fichier) pour charger les données suivantes dans les tables **THEME** et **TERRITOIRE** (INSERT).

THEME

- 1 Animaux
- 2 Sport
- 3 Bateaux
- 4 Ferme pédagogique
- 5 Parcs et jardins
- 6 Jeux pour enfants
- 7 Patrimoine
- 8 Parcours Sportifs
- 9 Golf
- 10 Sports nautiques
- 11 Parc d'attractions

TERRITOIRE

- 1 Autour du Louvres - Lens
- 2 Vallées & Marais
- 3 Côte d'opale

b) Tables **PARTICIPANT**, **INSCRIPTION**, **SITE**, **EVENEMENT**

Déterminer l'ordre d'insertion des données dans les tables et faire, au fur et à mesure des tests, les modifications si nécessaire pour que les insertions suivantes aboutissent.

- Table **PARTICIPANT** : Récupérer les données de la table `TESTS1.EMPRUNTEUR` (en premier) ainsi que les données des clients de la table `TESTS1.CLIENT`.
Utiliser une séquence pour générer les **cdPers** des différents participants insérés à partir de la table `TESTS1.CLIENT` et n'utiliser que les données utiles. Les données manquantes doivent être mises à `NULL`.
Les participants provenant de la table `TESTS1.CLIENT` sont considérés comme étant des particuliers (type personne = P)
- Table **RESERVATION** : récupérer les données de la table `TESTSAELD.INSCRIPTION` qu'il est possible de récupérer.
- Table **EVENEMENT** : récupérer les données de la table `TESTSAELD.EVENEMENT` qu'il est possible de récupérer.
- Table **SITE** : récupérer les données de la table `TESTSAELD.SITE` qu'il est possible de récupérer.

Ajouter quelques participants de votre choix grâce à l'interface de **SQL Developer**.

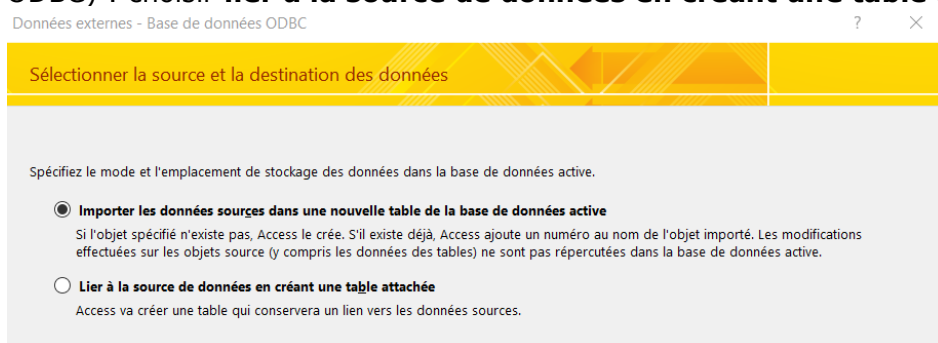
Colonnes Données Model Contraintes Droits Statistiques Déclencheurs Flashback Dépendances Détails Partitions Index SQL					
Trier... Filtre :					
	CDPERS	NOMPERS	PRENOMPERS	ADRPERS	CPP
1	Insérer une ligne (Ctrl+I)		Jean	12 bd de Reims	080
2	2	Durand	Pierre	51 bd Henry Vasnier	511
3	3	Taparov	Igor	101 rue Jean Jaures	083
4	4	Parant	Edith	10 rue des crayères	511
5	5	Balloir	Ivan	58 bis bd Gambetta	511
6	6	ODARI	Marine	17 rue Leon Blum	511
7	7	DIEUDONNE	Cesar	2 rue de champagne	511

c) Tables **PROGRAMME**

Pour faciliter la saisie des données de la table **PROGRAMME**, utiliser le **SBGD ACCESS**.

Pour cela :

- Créer une nouvelle base de données (tourisme62.accdb)
- Lier les tables **ACTIVITE**, **SITE** et **PROGRAMME** créées sur ORACLE (données externes/Nouvelles sources de Données/A partir d'autres sources/Bases de données ODBC) : choisir **lier à la source de données en créant une table attachée**



Choisir **bd19** dans l'onglet **Source de données machine**

Utiliser vos identifiants ORACLE pour vous connecter.

Dans la liste de tables qui s'affiche choisir vos tables **SITE**, **ACTIVITE** et **PROGRAMME** (elles sont préfixées de votre login).

- **Les activités**

Ajouter deux activités de votre choix à la liste des activités déjà proposées. Vérifier qu'elles sont bien visibles sous SQL DEVELOPPER.

- **Les activités au programme des sites**

Pour faciliter la saisie du programme des activités, il faut maintenant créer **un formulaire** qui permettra d'ajouter facilement des données dans la table **PROGRAMME**.

Le formulaire sera donc basé sur votre table **PROGRAMME** (propriété Source du formulaire).

Ajouter **une zone de liste déroulante** qui permettra de choisir un Site dans la liste des sites de la base.

Propriétés de la **zone de liste déroulante** pour les **Sites** :

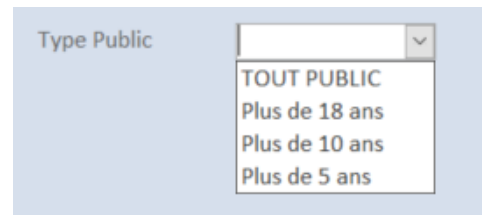
- Source contrôle : **CDSITE**
- Contenu : Requête qui sélectionne le code, le nom de tous les sites triés par ordre alphabétique
- Origine source : Table/Requête
- Colonne liée : 1
- Nombre de colonnes : 2
- Largeur colonnes : 0



De la même façon, créer une **zone de liste déroulante** pour les **activités triées**.

Créer une autre **zone de liste de liste déroulante** pour le **type de public** :

- Source contrôle : **TPPUBLIC**
- Contenu : "TOUS" ; "Tout Public" ; ...
- Origine source : Liste valeurs
- Colonne liée : 1
- Nombre de colonnes : 2
- Largeur colonnes : 0



Enfin, ajouter un bouton de commande qui permet de passer à un nouvel enregistrement (vide).

Ajouter

Utiliser le formulaire et vérifier l'ajout des données dans la table ORACLE (sous SQL Developer).



Utiliser la liste donnée en annexe pour saisir les Activités des sites.